


**Довідка
про відсутність плагіату у випускній кваліфікаційній роботі**

Міністерство освіти і науки України
Український державний університет науки та технологій
Кафедра «Комп'ютерні інформаційні технології»

ДОВІДКА

За результатами перевірки випускної кваліфікаційної роботи здобувача вищої освіти
Саєнка Є.О.
на тему: "Аналіз динаміки розвитку досліджень в галузі комп'ютерних наук на основі
бібліографічних посилань"
в роботі не виявлено порушень академічної доброчесності.

Керівник ВКР  Андрющенко В.О.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Український державний університет науки і технологій

Кафедра Комп'ютерні інформаційні технології

«ДО ЗАХИСТУ»

Завідувач кафедри

В.Г. /Вадим ГОРЯЧКІН/

« 20 » 12 2021 р.

ДИПЛОМНА РОБОТА

на здобуття освітнього ступеня «магістр»

Галузь знань 12 Інформаційні технології

Спеціальність 121 Інженерія програмного забезпечення

Тема Аналіз динаміки розвитку досліджень в галузі комп'ютерних наук на
основі бібліографічних посилань

Theme Analysis of the research dynamics in the computer science based on
bibliographic references

Керівник дипломної роботи

доц. В.А. Вадим АНДРЮЩЕНКО

Нормоконтролер

доц. О.К. Олена КУРОП'ЯТНИК

Студент групи ПЗ2021

Є.С. Євген САЄНКО

Student

Yevhen SAIENKO

Дніпро – 2021

Дніпровський національний університет залізничного транспорту імені
академіка В. Лазаряна
Факультет Комп'ютерних технологій і систем Кафедра Комп'ютерні
інформаційні технології
Спеціальність Інженерія програмного забезпечення

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

проф. Шинкаренко В.І.
(підпис)

«__» _____ 2020 р.

ЗАВДАННЯ

до дипломної роботи на здобуття ОС _____ Магістр _____

(освітній ступень)

студента групи (ПЗ2021) 961-М Саснка Євгена Олеговича
(номер групи) (ПІБ)

1 Тема дипломної роботи: Аналіз динаміки розвитку досліджень в галузі
комп'ютерних наук на основі бібліографічних посилань


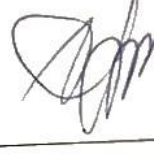
2 Термін подання студентом закінченого проекту «3» грудня 2021 р.

3 Вихідні дані до дипломного проекту _____

4 Зміст пояснювальної записки (перелік питань до розробки) Призначення,
постановка задачі та огляд аналогів, методика проведення дослідження,
проектування і розробка інструментального програмного забезпечення,
дослідження ефективності засобів синхронізації, охорона праці та безпека в
надзвичайних ситуаціях, висновки.

5 Перелік демонстраційного матеріалу Постановка задачі, опис аналогів, методи
вирішення, механізми реалізації, аналіз результатів, висновки.

6. Консультанти (з назвами розділів):

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Техніко-економічні розрахунки	доцент <u>Гненний М.В.</u>		
Охорона праці та безпека в надзвичайних ситуаціях	доцент <u>Саблін О.І.</u>		

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва розділів дипломного проекту	Термін виконання розділів проекту (роботи)	Примітка
1	Вступ	09.06.2021 – 15.06.2021	
2	Огляд літератури	17.06.2021 – 15.07.2021	
3	Постановка задачі, технічне завдання	15.08.2021 – 01.09.2021	
4	Створення тестової програми	15.09.2021 – 15.10.2021	
5	Перші тестування	15.10.2021 – 22.10.2021	30%
6	Аналіз результатів	30.10.2021 – 11.11.2021	
7	Розрахунок економічних показників	11.11.2021 – 14.11.2021	
8	Охорона праці	14.10.2021 – 18.10.2021	60%
9	Оформлення пояснювальної записки	19.10.2021 – 01.12.2021	
10	Демонстраційні матеріали	5.11.2021 – 03.12.2021	100%

Дата видачі завдання «18» листопада 2020 р.

Керівник дипломного проекту

(підпис)

В.О. Андрущенко

(ПБ)

Завдання прийняв до виконання

(підпис)

Є.О. Сасенко

(ПБ)

РЕФЕРАТ

Об'єктом роботи є напрямки досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

Предметом виступає динаміка розвитку напрямків досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

Для розв'язання поставлених завдань нами були використані такі методи дослідження: теоретико-критичний аналіз літератури з теми дослідження; зіставлення, узагальнення і синтезування здобутої інформації, статистичний аналіз на теорія планування експериментів.

Результати та їх новизна: дослідження робить внесок у аналіз динаміки розвитку досліджень в галузі комп'ютерних наук на основі знайдених бібліографічних посилань. Результати дослідження дозволяють зробити висновки щодо динаміки розвитку досліджень в галузі комп'ютерних наук. Пояснювальна записка складається зі вступу, 5 розділів, висновків, бібліографічного списку та 4 додатків:

- вступ описує актуальність обраної теми, складається із 3 сторінок;
- перший розділ складається з 24 сторінок і описує сучасний стан пошукових систем;
- другий розділ складається з 12 сторінок і містить обґрунтування експериментального методу дослідження;
- третій розділ складається з 26 сторінок, описує архітектуру системи;
- четвертий розділ складається з 14 сторінок де описані експерименти;
- п'ятий розділ розкриває питання охорони праці, складається з 7 сторінок.
- додатки містять технічне завдання, робочий проект та публікації.

Усього робота містить: таблиць – 21, рисунків – 54, бібліографія – 55 джерел.

Ключові слова: бібліографічні посилання, тексти.

ЗМІСТ

Вступ.....	12
1 Огляд проблеми.....	15
1.1 Призначення та сфера застосування.....	15
1.2 Постановка задачі.....	19
1.3 Аналіз предметної сфери.....	20
1.4 Огляд програмних аналогів.....	28
Висновки до розділу 1.....	36
2 Обґрунтування експериментального методу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.....	37
2.1 Часова ефективність алгоритму.....	37
2.2 Функціональна ефективність алгоритму.....	46
Висновки до розділу 2.....	49
3 Проектування й розробка інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.....	51
3.1 Формалізація задачі.....	51
3.2 Базова архітектура системи.....	58
3.3 Внутрішнє проектування.....	61
3.4 Розробка інтерфейсу користувача.....	63
3.5 Тестування та налагодження програми.....	67
Висновки до розділу 3.....	76
4 Дослідження ефективності системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.....	77
4.1 Підготовка до експерименту.....	77
4.2 Проведення експерименту.....	80

	10
4.3 Результати експерименту.....	81
Висновки до розділу 4	90
5 Охорона праці та безпека в надзвичайних ситуаціях.....	91
5.1 Вимоги безпеки при виконанні робіт на робочому місці	91
5.2 Шкідливі фактори на робочому місці.....	92
5.1.1 Характеристики робочого місця.....	93
5.1.2 Освітлення	93
5.1.3 Мікроклімат	94
5.1.4 Шум та вібрація.....	95
5.3 Дії працівників в надзвичайних ситуаціях	95
Висновки	98
Список використаної літератури	100
Технічне завдання	
Робочий проект	
Тезис	

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АСКР	автоматизована система комплексних розрахунків
АСР	автоматизована система розрахунку
БД	база даних
ЕОМ	електронно-обчислювальна машина
ОС	операційна система
ПЗ	програмне забезпечення
ПК	персональний комп'ютер
СУБД	система управління базою даних
СУД	система управління даними
ТУ	технічне управління

ВСТУП

Актуальність теми дослідження. На сьогоднішній день пошукові системи є складними та величезними механізмами, які являють собою не тільки інструмент для знаходження будь-якої необхідної інформації, але й досить цікаві сфери для бізнесу. Робота за допомогою пошукових систем дозволяє багатьом користувачам глобальної мережі здійснювати швидкий пошук потрібної інформації у найкоротші терміни. Ця робота є актуальною, за рахунок того, що пошукові системи вже довгий час є обов'язковою частиною Інтернету та життя суспільства

Швидке зростання інформації в мережі зробили його океаном найрізноманітніших даних, важливість яких зростає пропорційно їх обсягу. За оцінкою експертів обсяг інформації, що передається каналами Інтернет, подвоюється кожні півроку. Щодня в мережі з'являються мільйони нових документів, і природно, що без систем пошуку вони в переважній більшості залишилися б не затребуваними, взагалі не були б ким знайдені, і вся та величезна кількість інформації виявилася б нікому не потрібною. Виникла необхідність створення таких засобів, які б легко орієнтуватися в інформаційних ресурсах глобальних мереж, швидко і надійно знаходити потрібні відомості. В Інтернеті з'явилися спеціальні пошукові засоби. Ще кілька років тому була така думка: в Інтернеті є все, але знайти там нічого неможливо. Однак з появою і швидким розвитком пошукових каталогів, пошукових машин, і всіляких пошукових програм ситуація змінилася, і тепер в Мережі інформацію іноді можна знайти швидше, ніж у книзі, що лежить на столі. Найбільш популярним та застосовуваним способом пошуку в Інтернеті є використання пошукових систем. Пошукова система – портал, який здійснює пошук, збирання та сортування інформації в мережі Інтернет. Пошукові системи це інструмент, що дозволяє користувачеві глобальної мережі в найкоротші терміни знайти інформацію, що його цікавить. Першочергове завдання будь-якої пошукової системи – доставляти людям ту

інформацію, яку вони шукають. Отримуючи результат, користувач оцінює роботу системи, керуючись кількома основними параметрами (швидкість пошуку, точність пошуку та кількість результатів).

Мета та завдання дослідження. Метою даної дипломної роботи є здійснити аналіз динаміки розвитку досліджень в галузі комп'ютерних наук на основі знайдених бібліографічних посилань за допомогою розробленого програмного забезпечення.

Для досягнення поставленої мети у роботі необхідно виконати низку завдань:

- розкрити експериментальний метод досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, визначити його дієвість та ефективність;
- здійснитит проектування й розробку інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань;
- провести дослідження ефективності системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

Об'єкт та предмет дослідження. Об'єктом роботи є напрямки досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

Предметом виступає динаміка розвитку напрямків досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

Методи наукового дослідження. Методи дослідження включають методи теорії управління і прийняття рішень, штучного інтелекту, теорії ймовірності, методи експертного аналізу, інформатики.

Наукова новизна дослідження. Вперше розроблено структуру та алгоритми функціонування системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань у якій на відміну від інших є загальна структура програмного модулю визначення коефіцієнтів важливості, яка позиціонується виводом графічних результатів, в тому числі, за рахунок вперше застосованих в цих цілях методу моделювання складних

знань (рішень) і методу імовірнісного оцінювання складних знань, а також нового програмного блоку аналізу і вирішення характерних модельних ситуацій.

Ефект від впровадження дозволить спростити схему пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, що сприятиме економії часу та коштів.

Практична значимість дослідження. Практична значимість даної роботи полягає в тому, що результат досягнення поставленої мети має можливість застосування в діяльності конкретного закладу, а також може бути використаний і іншими закладами для пошуку тем досліджень в галузі комп'ютерних наук на основі бібліографічних посилань в умовах сучасного функціонування, також концепція викладення даного дослідження може бути використана в якості методичної допомоги при розробці інформаційних систем.

Апробація результатів. Основні положення магістерської роботи доповідалися та були схвалені на науковому семінарі кафедри КІТ (09.12.2021р.)

Структура роботи. Структуру роботи включають: вступ, п'ять розділів, висновки, список використаної літератури та додаток. Загальний обсяг роботи становить 192 сторінки.

1 ОГЛЯД ПРОБЛЕМИ

1.1 Призначення та сфера застосування

Будь-яке наукове дослідження передбачає отримання нових знань. Для цього важливо вміти працювати з науковою інформацією – насамперед науковими публікаціями, адже щоб зрозуміти новизну власних результатів треба дуже добре уявляти, що і як було зроблено попередниками. Кожна наукова спеціальність має свою специфіку, пов'язану як з тим, що, як і де публікують, так і з тим, хто як і надалі ці публікації використовує. Одна дуже важлива особливість того напрямку, полягає в тому, що фахівець в ідеалі повинен знати всі публікації за своєю тематикою незалежно від дати, місця та мови публікації. До речі, у тому числі й з цієї причини імпакт-фактори мало можуть сказати про рівень журналу, що спеціалізується на систематиці: цитованість робіт багато в чому визначається наявністю фахівців за тією чи іншою групою, а середній вік публікацій, на які посилається дослідник-систематик, зазвичай перевищує 50 років.

З кожним днем збільшується кількість авторів, журналів, навчальних закладів та електронних архівів, які дозволяють будь-якій людині отримати доступ до їхніх наукових статей через Інтернет. Щоб було легше орієнтуватися серед цієї розосередженої інформації, створюються електронні бібліотеки наукових статей. Чим більше статей включає бібліотека, тим корисніша вона користувачам. Однією з найважливіших проблем є своєчасне додавання до бібліотеки нових статей.

Поповнення електронної бібліотеки науковими статтями може здійснюватися такими способами:

1. Додавання нових наукових статей до бібліотеки вручну. Таким чином, поповнення може здійснюватись користувачами ЕБ, або самими авторами чи виданнями, які хочуть розмістити свої статті у бібліотеці.

2. Напівавтоматичний пошук нових статей. Так, у [1] описана процедура Relevance Feedback.

3. Автоматичний пошук наукових статей у мережі Інтернет та їхнє додавання до бібліотеки. Цей метод було запропоновано у роботі [2]. Він дозволяє охопити велику різноманітність статей, представлених у мережі Інтернет.

Застосування існуючих англомовних напрацювань для пошуку в україномовному сегменті мережі Інтернет (Укрнет) обмежено такими особливостями Укрнет:

- у Укрнеті відсутні такі якісні безкоштовні джерела бібліографічної інформації як DBLP;
- традиція американських учених вести свої домашні сторінки з публікаціями не поширена на території України. У той час, як існуючі підходи використовують пошук домашніх сторінок;
- часто домашня сторінка вченого (або сторінка організації) надає не тексти статей, а лише бібліографічні посилання;
- в Укрнеті багато публікацій доступні не у форматі PS, а в PDF, DOC, HTML. І якщо у США формат PS використовується в основному для наукових публікацій, що активно використовується в CiteSeer, то інші формати використовуються для публікацій різного призначення;
- у Укрнет немає сервісів типу HPSearch для пошуку домашніх сторінок дослідників;
- на відміну від англомовних статей, у яких існують загальноприйняті норми структурування та оформлення статті, для україномовних статей немає таких норм, і автори структурують та оформлюють статті, керуючись виключно своїми побажаннями (вимоги різних конференцій та журналів також дуже відрізняються);
- на відміну від англійської мови, де розділи статті мають традиційні назви (Abstract, Introduction тощо) в українській мові використовується

велика різноманітність слів для позначення одних і тих самих розділів (наприклад, лише бібліографія може називатися «література», «посилання», «джерела», «список літератури» і т.п.).

Завдання автоматичного поповнення електронної бібліотеки (ЕБ) полягає в тому, щоб автоматично знаходити в Інтернеті доступні статті, яких ще немає в її базі, і додавати їх до бібліотеки.

Були виділені такі окремі випадки автоматичного поповнення ЕБ:

1. Пошук тексту статті за заданими авторами та назвою. Потрібно знайти статтю, якої немає в бібліотеці, якщо відома назва цієї статті та її автори. Таке завдання може виникнути, наприклад, коли користувач сам шукає конкретну статтю, ставлячи її метаінформацію. Також у вже знайденій статті може міститися бібліографічне посилання на іншу статтю, якої ще немає у бібліотеці. У цьому випадку з бібліографічного посилання можна отримувати назву та авторів статті, що призводить до розглянутої задачі.

2. Пошук домашньої сторінки автора/конференції/ установи/видання, що містить список публікацій, за заданим ім'ям автора, назвою конференції/установи/видання. І пошук нових статей на цій сторінці. На домашніх сторінках вчених, сайтах різних конференцій, установ та видань часто містяться посилання різні наукові статті. Завдання полягає в тому, щоб знайти такі сторінки, розпізнати посилання на тексти наукових статей і додати нові статті в бібліотеку. Таке завдання виникає, коли є якісь статті одного автора, або які належать до однієї конференції/установи/видання, і хочеться знайти інші статті цього автора/конференції/установи/видання.

3. Пошук статей по заданим авторам та предметній області. Нехай є стаття певної теми. Можна припустити, що автори цієї статті займаються дослідженнями в області, до якої належить стаття, що розглядається. Тоді слід спробувати знайти інші статті, задаючи їхні імена та ключові слова, що описують предметну область.

4. Пошук статей із заданої предметної області. У різних тематичних спільнотах, блогах, на форумах, присвячених будь-яким науковим областям,

часто поділяються посилання на різні статті заданої тематики. Завдання полягає в тому, щоб знайти такі сторінки, витягти з них посилання на публікації та завантажити нові статті до бібліотеки.

5. Періодичний пошук нових статей у відомих джерелах. На домашніх сторінках вчених, сайтах конференцій/установ/видань час від часу з'являються нові статті, які також потрібно додавати до бібліотеки. Тому необхідно періодично здійснювати пошук оновлень у відомих джерелах наукових статей.

Можна виділити такі основні типи наукових публікацій, більшість з яких потребують своїх способів пошуку:

1. Статті в наукових журналах та інших серіальних виданнях (праці наукових товариств, інститутів тощо). Зараз це наймасовіший тип публікацій, у якому публікується найважливіша інформація. Але так було не завжди – років 30-40 тому дуже велике значення мало також написання монографій, ну а на зорі розвитку науки в монографіях публікувалася більшість наукових даних.

Як правило, практично всі наукові журнали в даний час розповсюджуються в електронному вигляді, але є деякі журнали, які досі доступні лише у друкованій версії.

Це найдоступніший тип публікацій. Дуже багато наукових журналів оцифровані цілком і доступні в режимі онлайн у тому чи іншому вигляді (на жаль, це невеликою мірою стосується вітчизняних видань).

2. Монографії та збірники статей. До цього часу в описових науках монографії залишаються найважливішим джерелом як нових даних, і великих узагальнень.

На відміну від журналів, істотна частина книг досі поширюється лише у друкованому вигляді. Дещо кращі справи з серіальними виданнями, але і тут поки що ситуація гірша ніж з журналами.

Більш-менш доступні порівняно свіжі або, навпаки, старі роботи (написані 100 років тому і більше).

3. Тези доповідей та матеріали конференцій. Як правило, публікації, що не рецензуються і не редагуються, в яких, однак, іноді вперше з'являються важливі відомості, лише потім одержують більш-менш повний вираз у статтях.

В останні роки 10 подібні публікації, як правило, в тому чи іншому вигляді розміщуються в Інтернеті, але шанси знайти в мережі тези якоїсь конференції 80-х років близькі до нуля.

4. Дисертації. Це абсолютно унікальне джерело даних, тому що багато відомостей присутні тільки в дисертаціях, а потім з тих чи інших причин не доводяться до публікацій або публікуються у сильно скороченому обсязі.

1.2 Постановка задачі

Метою даної дипломної роботи є здійснити аналіз динаміки розвитку досліджень в галузі комп'ютерних наук на основі знайдених бібліографічних посилань за допомогою розробленого програмного забезпечення.

Для досягнення поставленої мети у роботі необхідно виконати низку завдань:

- розкрити експериментальний метод досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, визначити його дієвість та ефективність;
- здійснити проектування й розробку інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань;
- провести дослідження ефективності системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

1.3 Аналіз предметної сфери

Більш менш серйозний підхід до будь-якої задачі починається з аналізу можливих методів її вирішення. Пошук інформації в Інтернеті може бути проведений декількома методами, що значно розрізняються як за ефективністю та якістю пошуку, так і за типом видобутої інформації. У ряді випадків доводиться використовувати дуже трудомісткі методи.

Можна виділити такі основні методи пошуку інформації в Інтернеті, які, залежно від цілей та завдань того, хто шукає, використовуються окремо або в комбінації один з одним:

- безпосередній пошук з використанням гіпертекстових посилань;
- використання пошукових машин;
- пошук із застосуванням спеціальних засобів;
- аналіз нових ресурсів;
- безпосередній пошук з використанням гіпертекстових посилань.

Оскільки всі сайти у просторі WWW фактично виявляються пов'язаними між собою, пошук інформації може бути здійснений шляхом послідовного перегляду пов'язаних сторінок за допомогою браузера. Хоча цей повністю ручний метод пошуку виглядає повним анахронізмом в Мережі, що містить сотні мільйонів вузлів, "ручний" перегляд Web-сторінок часто виявляється єдиним можливим на заключних етапах інформаційного пошуку, коли механічне "копання" поступається місцем глибшого аналізу. Використання каталогів, класифікованих та тематичних списків та різноманітних невеликих довідників також відноситься до цього виду пошуку.

Використання пошукових машин.

Сьогодні цей метод є одним з основних та фактично єдиним при проведенні попереднього пошуку. Результатом останнього може бути перелік ресурсів Мережі, які підлягають детальному розгляду. Як правило, застосування пошукових машин засноване на використанні ключових слів, які передаються пошуковим серверам як аргументи пошуку: що шукати.

Якщо робити все правильно, формування списку ключових слів вимагає попередньої роботи зі складання тезауруса.

Пошук із застосуванням спеціальних засобів.

Цей повністю автоматизований метод може бути дуже ефективним щодо первинного пошуку. Одна з технологій цього методу заснована на застосуванні спеціалізованих програм – спайдерів, які в автоматичному режимі переглядають Web-сторінки, відшукуючи на них потрібну інформацію. Фактично це автоматизований варіант перегляду за допомогою гіпертекстових посилань, описаний вище (пошукові машини для побудови своїх індексних таблиць використовують схожі методи). Немає потреби говорити, що результати автоматичного пошуку обов'язково вимагають подальшої обробки. Застосування даного методу є доцільним, якщо використання пошукових машин не може дати необхідних результатів (наприклад, через нестандартність запиту, який не може бути адекватно заданий існуючими засобами пошукових машин). У ряді випадків цей метод може бути дуже ефективним. Вибір між використанням спайдера або пошукових серверів є варіантом класичного вибору між застосуванням універсальних або спеціалізованих засобів.

Аналіз нових ресурсів.

Пошук за новоствореними ресурсами може бути необхідним під час проведення повторних циклів пошуку, пошуку найсвіжішої інформації, або для аналізу тенденцій розвитку об'єкта дослідження у поступовій динаміці. Іншою можливою причиною може з'явитися те, що більшість пошукових машин оновлює свої індекси зі значною затримкою, викликаною гігантськими обсягами оброблюваних даних, і ця затримка зазвичай тим більше, чим менш популярна цікава тема. Це міркування може бути дуже істотним під час проведення пошуку у вузькоспеціальній предметній області.

Технологія пошуку за допомогою пошукових машин. Визначення географічних регіонів пошуку.

Розглянемо основні етапи, які однак присутні при пошуку інформації з використанням пошукових машин.

Оскільки проведення інформаційного пошуку переслідує практичні цілі – маркетингові, виробничі, суто утилітарні тощо – практична цінність інформаційного ресурсу може залежати і від географічного розташування відповідного джерела.

Складання тезаурусу.

Для використання пошукових серверів необхідний список ключових слів, організований з урахуванням семантичних відносин між ними, тобто. тезаурус. При складанні тезаурус необхідно передбачити обробку синонімів, омонімів і морфологічних варіацій ключових слів.

Використання законів Зіпфа.

Число, що показує скільки разів зустрічається слово у тексті, називається частотою входження слова. Якщо розташувати частоти в міру спадання та пронумерувати, то порядковий номер частоти називається рангом частоти. Імовірність виявлення слова в тексті = частота входження слова/число слів у тексті. Зіпф виявив, що якщо помножити ймовірність виявлення слова в тексті на ранг частоти, то величина, що вийшла, приблизно постійна для всіх текстів однією мовою:

$$C = (\text{Частота входження слів } X \text{ ранг частоти}) / \text{число слів}.$$

Це означає, що графік залежності рангу від частоти – рівностороння гіпербола. Зіпф також встановив, що залежність кількості слів з частотою від частоти - також гіпербола і постійна всім текстів у межах однієї мови.

Дослідження вищезазначених залежностей для різних текстів показали, що найбільш значущі слова тексту лежать у середній частині діаграми, оскільки слова з максимальною частотою зазвичай є прийменниками, частинками, займенниками, в англійській мові – артиклями (так звані "стоп-слова"), а рідко зустрічаються слова найчастіше немає вирішального

значення. Ґрунтуючись на цій закономірності, можна запропонувати таку методику.

Складання списку ключових слів.

Правильний набір ключових слів має значення для оптимального пошуку інформації. Наприклад, задавши пошуковій машині як ключове слово "АРП", отримаємо список документів, у яких зустрічається ця аббревіатура (Агентство з розвитку підприємства). Але якщо нас цікавлять документи на ширшу тему, наприклад: розвиток підприємництва, і ми сформуємо простий запит із цих двох слів, то пошукова машина видасть нам список із сотень тисяч найменувань, орієнтуватися в якому буде дуже непросто.

Тому для складання оптимального набору ключових слів використовують процедуру, засновану на застосуванні законів Зіпфа, яка полягає в наступному: беруть будь-яке текст-джерело, близьке до шуканої теми, тобто "зразок", і аналізують його, виділяючи значущі слова. У якості текст-джерела може бути книга, стаття, Web-сторінка, будь-який інший документ.

Аналіз тексту проводиться таким чином:

1. Видалення з тексту стоп-слів.
2. Обчислення частоти входження кожного слова та складання списку, в якому слова розташовані в порядку зменшення їх частоти.
3. Вибір діапазону частот, що лежить у середині списку, і відбір із цього діапазону слів, які найбільш повно відповідають змісту тексту.
4. Складання запиту до пошукової машини у формі перерахування відібраних таким чином ключових слів, пов'язаних з логічним оператором АБО (OR). Запит у такому вигляді дозволяє виявити тексти, в яких зустрічається хоча б одне з цих слів.

Число документів, отриманих в результаті пошуку за цим запитом, може бути величезним. Однак, завдяки ранжуванню документів (розташуванню їх у порядку зменшення частоти входження слів запиту в документ), що застосовується в більшості пошукових машин, на перших сторінках списку практично всі документи виявляються релевантними, причому документ-джерело може бути далеко від початку.

Більш адекватною представляється структура тезауруса у вигляді так званих семантичних зрізів, де для кожного основного терміну окремо будується таблиця супутніх слів і шумових слів (які не повинні зустрічатися в джерелі), - деякі пошукові машини (AltaVista) дозволяють це використовувати. Таким чином замість єдиної ієрархічної структури термінів отримуємо пакет таблиць, які можуть розширюватися і модифікуватися окремо.

Відбір пошукових машин.

Встановлюється послідовність використання пошукових машин відповідно до зменшення очікуваної ефективності пошуку із застосуванням кожної машини. Всього відомо близько 180 пошукових серверів, що розрізняються по регіонах охоплення, принципам проведення пошуку (а отже, за вхідною мовою та характером запитів, що сприймаються), обсягом індексної бази, швидкості оновлення інформації, здатністю шукати "нестандартну" інформацію тощо. Основними критеріями вибору пошукових серверів є обсяг індексної бази сервера і рівень розвиненості самої пошукової машини, тобто рівень складності сприйманих нею запитів.

Складання та виконання запитів до пошукових машин.

Це найбільш складний та трудомісткий етап, пов'язаний з обробкою великої кількості інформації (переважно шумової). На основі тезауруса формуються запити до вибраних пошукових серверів, після чого можливе уточнення запиту з метою відсічення явно нерелевантної інформації. Потім проводиться відбір ресурсів, починаючи з найцікавіших, з погляду цілей

пошуку. Дані з ресурсів, визнаних релевантними, збираються для подальшого аналізу.

Формування запитів.

Як формат, так і семантика запитів варіюються в залежності від застосовуваної пошукової машини та конкретної предметної області. Запити складаються так, щоб область пошуку була максимально конкретизована та звужена. Перевага надається використанню кількох вузьких запитів у порівнянні з одним розширеним. У випадку для кожного основного поняття з тезауруса готується окремий пакет запитів. Так само проводиться пробна реалізація запитів – як для уточнення та поповнення тезаурусу, так і з метою відсікання шумової інформації.

Мови запиту різних машин пошуку в основному є поєднанням наступних функцій:

Оператори булевої алгебри AND, OR, NOT:(I) – здійснюється пошук документів, утримуючих всі терміни, з'єднані даним оператором; (АБО) – шуканий текст повинен містити хоча б один з термінів, з'єднаних даним оператором; (НЕ) – пошук документів, в тексті яких відсутні терміни, що йдуть за даним оператором.

Оператори відстані – обмежують порядок проходження та відстані між словами, наприклад: – другий термін повинен знаходитися на відстані від першого, що не перевищує певної кількості слів; ВУ – терміни слідують у заданому порядку; – терміни, з'єднані оператором, є суміжними.

Можливість усічення термінів – використання символу "*" замість закінчення терміну дозволяє включити до списку всі слова, похідні від його початкової частини (шаблону).

Облік морфології мови – машина автоматично враховує всі форми даного терміна, можливі у мові, якою ведеться пошук.

Можливість пошуку за словосполученням, фразою.

Обмеження пошуку елементом документа (слова запиту мають знаходитися саме у заголовку, першому абзаці, посиланнях тощо).

Обмеження за датою опублікування документа.

Обмеження кількості збігів термінів.

Можливість пошуку графічних зображень.

Чутливість до малих і великих літер.

Результат запиту (список посилань) обробляється у два етапи. На першому етапі проводиться відсікання очевидно нерелевантних джерел, що потрапили у вибірку через недосконалість пошукової машини або недостатню "інтелектуальність" запиту. Паралельно проводиться семантичний аналіз, що має на меті уточнення тезаурусу для модифікації наступних запитів. Подальша обробка проводиться шляхом послідовного звернення на кожен із знайдених ресурсів та аналізу інформації, що там знаходиться.

Одна із проблем, що виникають у процесі пошуку є суто методологічною. Для проведення ефективного пошуку користувач зацікавлений в одночасному вирішенні двох протилежних завдань:

- збільшення охоплення з метою отримання максимальної кількості значущої інформації;
- зменшення охоплення з метою мінімізації шумової інформації.

Неважко бачити, що одночасно здійснити це досить складно, хоча найчастіше можливо. Один із методів, якщо пошукова машина дозволяє, - це запровадження явних обмежень (заборонених слів). Інший полягає у правильному формуванні запитів, зокрема, на перевагу кількох конкретизованих запитів одному загальному. На жаль, дуже обмежена вхідна мова більшості машин не залишає особливого простору для творчості у цьому напрямку.

Інша проблема – багатоваріантність людської мови. Якщо в англійській деякі слова мають безліч різних значень, то українська відрізняється багатством морфологічних варіацій слів, а для повноти пошуку необхідно враховувати ще й синоніми.

Часто у сфері українського Інтернету виникають суто технічні труднощі через різні кодування інформації. Українські пошукові машини розпізнають кодування користувача та шуканого сайту, але поєднати їх вдається не завжди.

Ще одна особливість українськомовної частини мережі – її нестабільність. Постійно змінюються адреси та структура сайтів, вони з'являються і зникають, і пошукові машини не встигають оновлювати свої бази індексованих даних, тому значна частина списку документів, виданого машиною, може бути недоступною [3].

Основною проблемою пошуку можна назвати невміння користувача ефективно шукати інформацію в мережі.

Пошукові машини шукають інформацію на всій мережі Internet. Насправді це не зовсім правильно. Якби при реалізації алгоритму роботи пошукових машин був використаний такий підхід, то для обробки лише одного запиту та видачі результатів знадобилося б кілька днів.

Кожна пошукова машина має та постійно поповнює свою (локальну) базу даних. База даних пошукової машини містить основні параметри (індекси) кожного відомого даній машині (проіндексованого) документа. Кожна пошукова машина використовує методи індексації. Крім того, різні пошукові системи мають різні обсяги бази даних.

В результаті, механізм обробки запиту користувача пошуковою машиною виглядає так:

- відповідно до заданого в запиті ключового слова або словосполучення, машина проводить пошук у своїй локальній базі даних, звіряючи ключове слово з наборами ключових слів, що відповідають кожному документу з її бази даних;

- потім, використовуючи відповідні алгоритми, пошукова машина сортує результати пошуку та видає їх користувачеві;

– в результаті сортування результатів, на початок списку поміщаються найбільш відповідні (з погляду пошуків ой машини) ключовими словами документи.

У зв'язку з великою кількістю інформації, розміщеної в мережі, жодна з пошукових машин не в змозі переглянути всі документи. Кожна пошукова машина індексує лише частину їх. Всі інші документи, а на жаль це більшість ресурсів, знайти з її допомогою не вдасться.

1.4 Огляд програмних аналогів

У загальному випадку результати пошуку насамперед залежать від поставленого завдання та коректності запиту. Але ці результати найчастіше, з одного боку, надмірні та з іншого боку неповні.

Проте, і автори та видавці, як правило, зацікавлені в тому, щоб інформація про публікації індексувалася пошуковими системами, але тут є нюанси: не завжди дозволяється індексація вмісту pdf-файлів, і в деяких випадках дозволена індексація сайтів лише певними пошуковими системами.

Окрім іншого, результати запиту залежать від порядку слів та від IP-адреси, з якої здійснюється пошук.

Якщо говорити про пошук публікацій, то питання «якою пошуковою системою користуватися» має одну відповідь – Google (це якщо не брати до уваги спеціалізовані бібліографічні пошукові системи).

По-перше, google досить повно індексує вміст Мережі. По-друге, велика кількість налаштувань розширеного пошуку (в т.ч. з використанням операторів) сильно полегшують роботу. По-третє, вміст пдф-файлів googl'ом індексується навіть у тому випадку, коли пдф складається із зображень і текстовий шар у файлі відсутній.

Рисунок 1.1 – Пошук google

Для пошуку публікацій найбільш корисними є розширені налаштування та оператори, що дозволяють обмежувати пошук файлами певного формату (наприклад, pdf за допомогою filetype:pdf), певними сайтами/доменами. Якщо шукається якась конкретна публікація, то бажано частину її назви або всю назву взяти в лапки.

Гугл має два окремі проекти, які мають пряме відношення до пошуку публікацій:

1) Google books – це фактично окрема пошукова система, що індексує вміст величезної кількості книг, журналів, збірників та інших видань. При цьому істотна частина публікацій доступна для завантаження у вигляді pdf (як правило, це старі видання, від початку XX століття і старше); залежно від IP список доступних для скачування видань може істотно відрізнятись, максимальна кількість робіт доступна користувачам США.

Досить багато публікацій доступні для перегляду повністю або частково. Такі роботи можна завантажити за допомогою спеціальних програм типу *EDS Google Book downloader* або плагінів (таких як *Greasemonkey* для

Mozilla у поєднанні з програмою для автоматичного завантаження файлів, наприклад *Download Master*).

І, нарешті, чималу користь можна отримати навіть від тієї інформації, яка є в публікаціях, які взагалі недоступні для перегляду в будь-якому вигляді крім фрагментів в кілька рядків (*snippet view*). З такими публікаціями, щоправда, є дві основні складності:

а) можна, звичайно, спробувати пошукати такі роботи десь ще, але ймовірність того, що з ними можна буде ознайомитися тільки в бібліотеці, досить велика.

б) у назвах джерел (особливо тих, які початково дані не латиницею) плутанини дуже і дуже багато, і інформація, що відображається зазвичай неповна.

Проте інформація, що міститься в таких фрагментах, може бути дуже важливою і практично не знаходитися іншими способами.

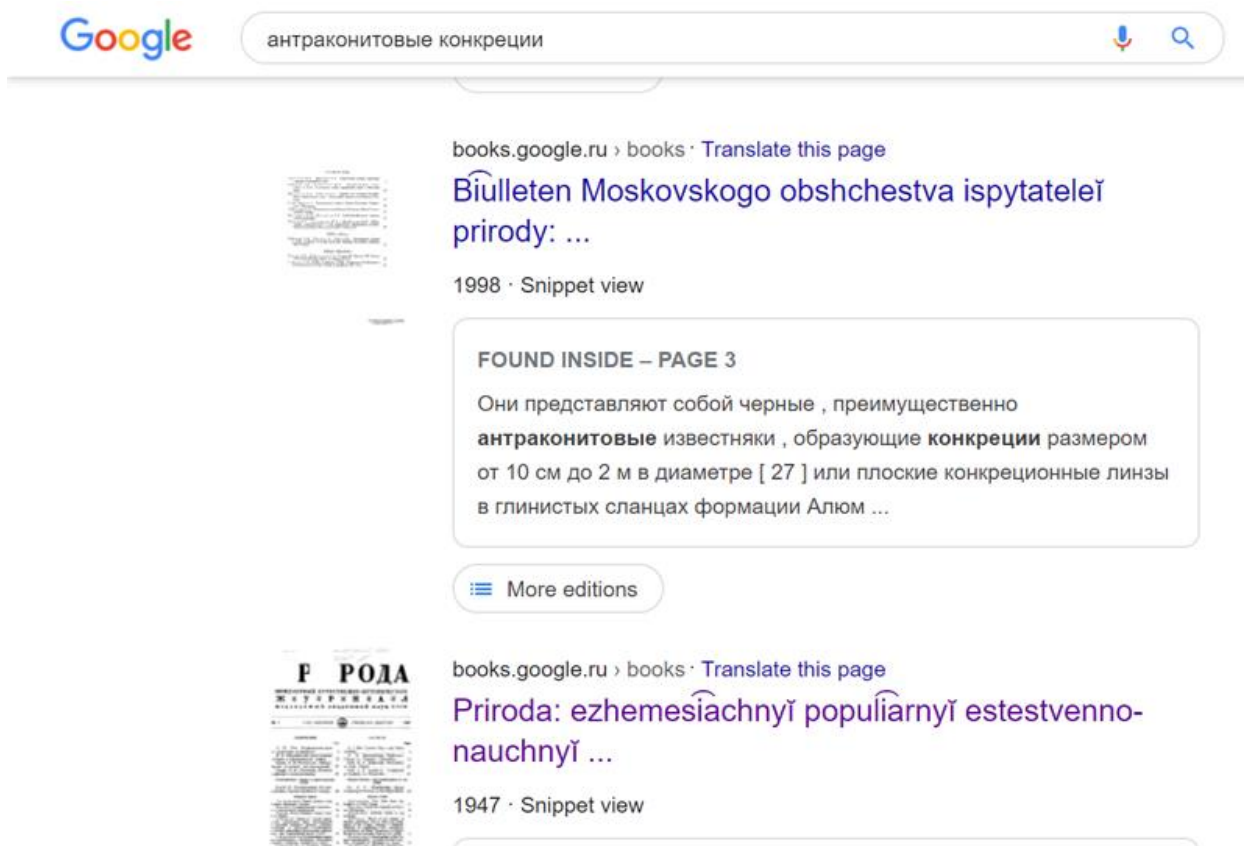


Рисунок 1.2 – Пошук Google books

2) Google Scholar. Це бібліографічна пошукова система, яка непогано шукає як самі статті, так і посилання на них, одночасно дозволяючи відразу скопіювати назви публікацій, відформатовані згідно з популярними типами цитування (APA, Harvard, ГОСТ, ДСТУ і т.д.). До зручностей даної системи варто віднести те, що індексуються не тільки сайти видавців, але і спеціалізовані соціальні мережі і різні сайти, де нерідко безплатно викладаються наукові роботи, і всі посилання на повнотекстові версії групуються в єдиний кластер. Тим не менш, Google Scholar індексує не всі публікації – це легко перевірити за допомогою ідентичного пошукового запиту *ключові слова filetype:pdf* Google та Google Scholar. Особливо яскраво ця різниця проявляється з ключовими словами, що рідко зустрічаються.

Ну а найбільш корисна функція google scholar – це можливість підписки на різні оповіщення.

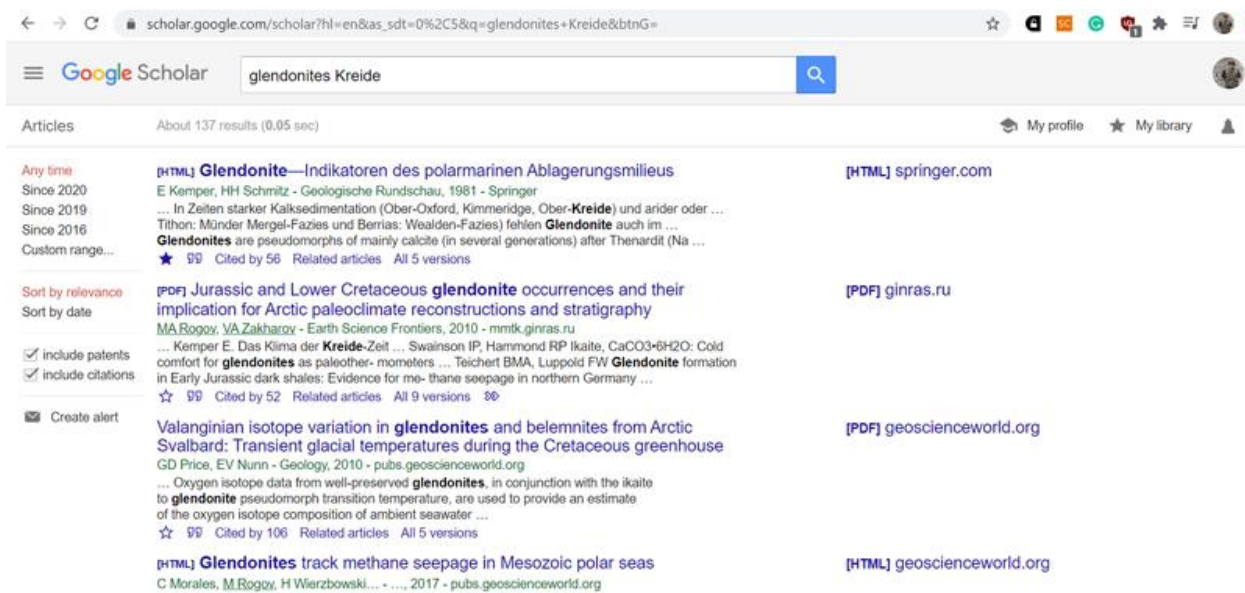


Рисунок 1.3 – Пошук Google Scholar

Бібліографічні пошукові системи (БПС), орієнтовані працювати з публікаціями, зараз дуже різноманітні і численні. Крім перерахованих вище

проектів Google можна відзначити такі сайти, які можуть розглядатися як БПС:

1) сайти, що індексують величезну кількість публікацій у всьому світі. В першу чергу це Scopus і Web of Science, доступні за підпискою (у випадку зі Scopus доступ також надається рецензентам Elsevier'овських журналів), а також найбільший сайт, що привласнює DOI публікаціям (CrossRef) або агрегатор інформації про публікації, гранти, дослідники тощо Dimensions.

Всі вони окрім Dimensions дозволяють шукати інформацію щодо обмеженого масиву даних – це переважно назва / ключові слова / резюме. У гірший бік тут виділяється CrossRef – там пошук іде тільки за назвою, причому зі строгою прив'язкою до форми слова. Щоправда, у CrossRef істотно більше проіндексовано російськомовних та україномовних публікацій, ніж у інших БПС із цього пункту, і плюс до того це найбільш зручний спосіб вирішити завдання типу «у мене є назва публікації, треба знайти її DOI».

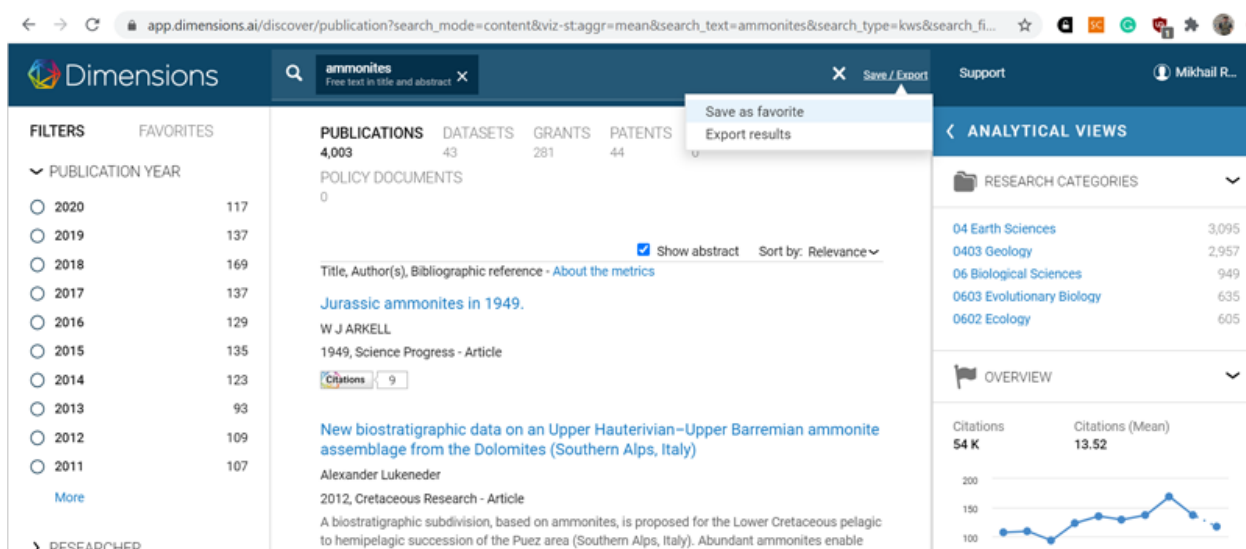


Рисунок 1.4 – Пошук у Dimensions

Dimensions – дуже цікавий проект, що недавно з'явився, в першу чергу завдяки безлічі найрізноманітніших налаштувань, широкому охопленню публікацій (індексуються тільки публікації з DOI, їх поки що трохи менше,

ніж є на CrossRef) і повнотекстовому пошуку. Вірніше, тут можна вибирати різні опції пошуку (повнотекстовий / за резюме / за назвою та ключовими словами). Результати можна сортувати найрізноманітнішими способами (дата / релевантність / число посилань / число альтметрик), і обмежувати за різними параметрами (джерело / автор / роки / тематика та багато іншого). У Dimensions є різні версії (включаючи платну та корпоративну), тут розглядається лише безкоштовний варіант. Окремо можна шукати інформацію як за публікаціями, так і за базами даних та грантами (остання опція доступна тільки за підпискою).

The screenshot shows the Dimensions web application interface. The search query is 'Kimmeridgian ammonites'. The left sidebar has a 'FILTERS' section with 'PUBLICATION YEAR' ranging from 2011 to 2020. The main content area has tabs for 'ANALYTICAL VIEWS' and 'PUBLICATIONS'. The 'PUBLICATIONS' tab is active, showing a 'Researchers' section related to the search. Below this, there is a table with columns: Name, Organization, Country, Publications, Citations, and Citations mean. The table lists three researchers: FEDER Ico Olóriz, Mikhail A Rogov, and Ana Bertha Villaseñor.

Name	Organization, Country	Publications	Citations	Citations mean
FEDER Ico Olóriz	University of Granada, Spain	14	151	10.79
Mikhail A Rogov	Geological Institute, Russia	9	142	15.78
Ana Bertha Villaseñor	National Autonomous University of Mexico, Mexi...	8	61	7.63

Рисунок 1.5 – Пошук у Dimensions

У вкладці Analytical view можна подивитися, наприклад, хто або в яких журналах публікувався за цікавою для нас тематикою в той чи інший час (в даному випадку – з 2016 по 2020 роки). Натиснувши на прізвище автора, можна подивитися з ким він публікувався, в яких журналах і т.д.

Додаткові опції пропонуються у вкладці Analytical view. Вони дозволяють легко зрозуміти, хто зараз або в будь-якому обраному часовому діапазоні займається тією чи іншою тематикою, в які журнали ці люди пишуть статті та з якими співавторами. Це зручний спосіб для пошуку

потенційних співавторів і рецензентів, особливо для тих, хто тільки почав займатися якоюсь тематикою і не дуже добре собі уявляє, що з нею у світовому масштабі робиться. Для тих дослідників, у яких у статтях є ORCID, у профілі наводиться і цей ідентифікатор, і Scopus author ID, а також (за наявності) ResearcherID/профіль на Publons, що чіпляється до них «автоматом». Dimensions – це вкрай корисний проект, причому інтуїтивно зрозумілий.

2) також як спеціалізовані БПС можна розглядати сайти найбільших міжнародних видавців (Elsevier, Wiley, Springer, Taylor & Francis і т.д.) та розповсюджувачів (Ingentaconnect, GeoscienceWorld) наукових видань. Втім, обмеження результатів пошуку тим чи іншим видавцем чи розповсюджувачем на користь, як правило, не йде і швидше може бути корисним для того, щоб коротко ознайомитися з тією чи іншою темою.

3) певною мірою функції БПС виконують наукові соціальні мережі (Academia.edu, ResearchGate), а також «гібрид» соціальної мережі та бібліографічного менеджера Mendeley (доступна як офлайн-версія у вигляді програми, так і її онлайн-варіант; зараз, після покупки Mendeley компанією Elsevier там доступні багато опцій Scopus). Втім, вміст наукових соціальних мереж добре індексується googl'ом, і тут хіба що є сенс регулярно переглядати стрічку оновлень у пошуках чогось зовсім нового.

4) в окрему категорію БПС можна виділити регіональні або спеціалізовані сайти, де в основному є дані про публікації, видані в будь-якій країні або кількох країнах, а також спеціалізовані сайти, присвячені якимось конкретним науковим напрямкам.

Характерною особливістю таких порталів є те, що вони вкрай неохоче дають індексувати свій вміст стороннім пошукачам, так що якщо потрібно знайти щось французьке або японське – надійніше заглянути на відповідні сайти та пошукати там.

Окремо слід зазначити про BHL. Це дуже корисний проект для всіх дослідників, які так чи інакше пов'язані з вивченням сучасних організмів.

Дану бібліотеку відрізняє широке охоплення джерел (включаючи різні рідкості) та наявність спеціальних пошукових інструментів (таких як пошук у вкладці Advanced search). Нерідко текстовий шар може бути розпізнаний неправильно (з помилково обраною мовою), а також жахлива якість стандартних ілюстрацій (якість поганого розмитого .djvu).

Оскільки для таксономічних досліджень якість зображень зазвичай має велике значення, то тут найбільш правильним підходом є завантаження потрібної публікації у форматі jp2, а потім – обробка файлів (спочатку переформатування у звичайний jpg/tiff, потім обробка ScanTailor та OCR). До речі, всі публікації з BHL розміщуються на archive.org, і іноді зручніше проводити повнотекстовий пошук саме по archive.org.

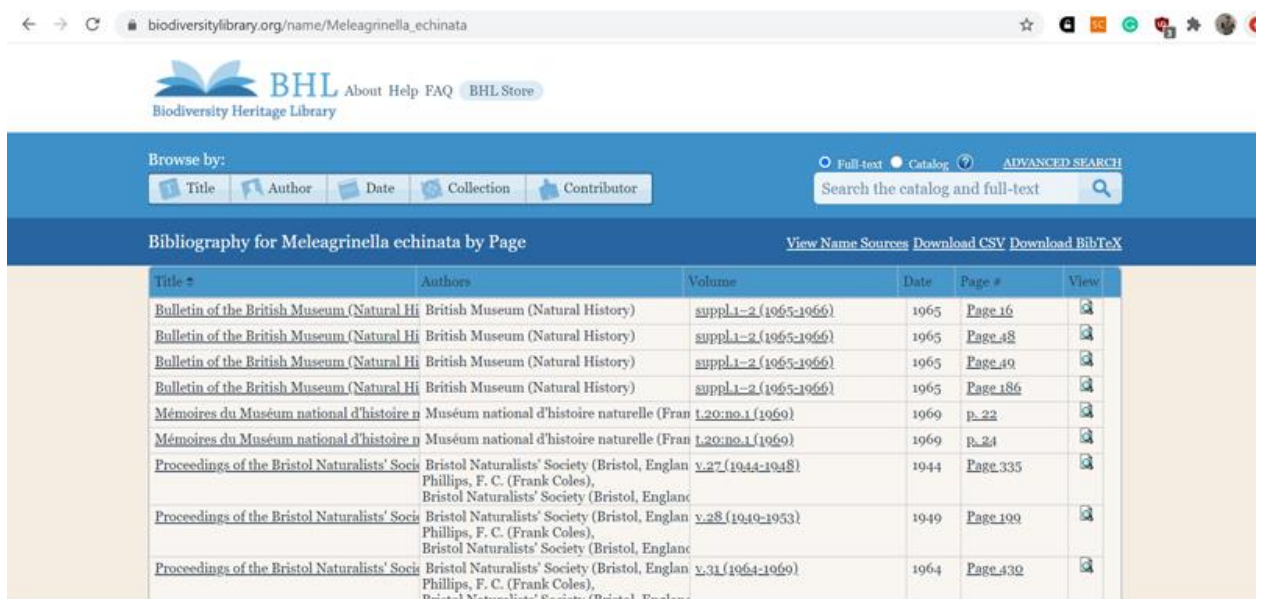


Рисунок 1.6 – Приклад видачі при пошуку за таксоном на BHL

5) У якості БПС можна розглядати і найбільші «піратські» проекти, що забезпечують вільний доступ до наукових публікацій – SciHub та LibGen, оскільки на них у тому чи іншому вигляді реалізована можливість пошуку за назвою публікації або ключовими словами

І якщо sci-hub може бути скоріше використаний як зручне доповнення до пошуку на Dimensions, то на LibGen регулярно з'являються рідкісні

монографії, яких в інших місцях немає – вони скануються ентузіастами та розміщуються на ЛібГені в приватному порядку.

І насамкінець окремо варто сказати про пошук дисертацій. Хоча багато дисертацій (як сучасні українські, так і іноді досить старі закордонні) викладені в Інтернеті у відкритому доступі та індексуються пошуковими системами, для отримання інформації про свіжі дисертації, які тільки планується захистити, має сенс заглядати на сайт ВАКу. Там зараз дисертації можна шукати за спеціальностями, ключовими словами, датою захисту та іншими параметрами (при цьому окремо пошук ведеться за ВАКівськими дисертаціями, а окремо – за тими, що захищаються у радах організацій, які мають право самостійного присудження ступенів).

Висновки до розділу 1

У межах першого розділу здійснено дослідження предметної області за темою дослідження, сформовано мету дослідження та визначено низку завдань виконання яких направлено на реалізацію мети.

Пошук інформації в мережі – це досить специфічна та кропітка робота, яка потребує певних знань та навичок. Для проведення пошукових робіт знадобляться: знання основних інформаційних ресурсів і вміння добре в них орієнтуватися, практичні навички роботи.

2 ОБҐРУНТУВАННЯ ЕКСПЕРИМЕНТАЛЬНОГО МЕТОДУ ДОСЛІДЖЕНЬ В ГАЛУЗІ КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

2.1 Часова ефективність алгоритму

У світі, Інтернет вже давно використовують як широкий довідковий інструмент. За останні роки він став середовищем обробки та зберігання наукової, ділової та інших типів інформації. Але основними особливостями Інтернету є динаміка інформації, її постійне оновлення та розповсюдження по всьому Інтернету.

Саме тому пошукові системи вже давно стали невід'ємною частиною Інтернету. Завдяки ним користувачі Всесвітнього павутиння намагаються знайти потрібну інформацію. Переважно пошук інформації зводиться до пошуку сторінки, на якій розміщена ця інформація, і таких сторінок в Інтернеті може бути декілька. Для отримання результатів, які дозволять порівняти інформацію на однотипних веб-ресурсах, необхідно створити тематичні пошукові системи.

Створення тематичної пошукової системи містить у своїй основі такі завдання:

- підбір тематичних сайтів,
- пошук інформації на різних ресурсах із різними запитами,
- вилучення інформації з сайту,
- створення єдиної системи надання інформації користувачеві.

Сервіс має веб-інтерфейс, у якому користувач вказує ключові слова та параметри для пошуку. Ці дані зберігаються в часовій таблиці, що створюється під кожен пошуковий запит користувача.

Вхідними даними для роботи алгоритму пошуку розробленого веб-сервісу є пошукові параметри, що вносяться в таблицю користувачем.

Алгоритм ґрунтується на стеммінгу тексту, з метою отримання більш точних результатів пошуку. Він виконує пошук основи слова, що враховує

морфологію вихідного слова. Виконуючи морфологічний розбір слова, знаходить загальну всім його граматичних форм основу, відсікаючи суфікси та закінчення.

Враховуючи поставлені завдання, алгоритм роботи тематичного пошукача схожий на алгоритм звичайної пошукової системи.

Пошукова система складається з таких основних компонентів:

Spider (павук) – браузероподібна програма, яка завантажує веб-сторінки.

Crawler (краулер, мандрівний павук) – програма, яка автоматично проходить за всіма посиланнями, знайденими на сторінці.

Indexer (індексатор) – програма, яка аналізує веб-сторінки, завантажені павуками.

Database (база даних) – сховище завантажених та оброблених сторінок.

Search engine results engine (система видачі результатів) – отримує результати пошуку з бази даних.

Web server (веб-сервер) – веб-сервер, який здійснює взаємодію між користувачем та іншими компонентами пошукової системи.

Web server та Search engine results engine часто називають просто пошуковим сервером.

У деталях реалізації пошукових механізмів можуть відрізнятися один від одного (наприклад, зв'язка Spider + Crawler + Indexer може бути виконана у вигляді єдиної програми, яка завантажує відомі веб-сторінки, аналізує їх та шукає за посиланнями нові ресурси), проте всім пошуковим системам властиві описані загальні риси.

Spider. Забезпечує завантаження сторінки та витягує всі внутрішні посилання з цієї сторінки. Завантажується HTML-код кожної сторінки Для завантаження сторінок роботи використовують протоколи HTTP. Працює "павук" так. Робот передає на сервер запит "get/path/document" та деякі інші команди HTTP-запиту. У відповідь робот отримує текстовий потік, що містить службову інформацію і сам документ.

Посилання вилучаються з тегів a, area, base, frame, frameset та ін. Крім посилань, роботи обробляють редиректи (перенапрявлення). Кожна завантажена сторінка зберігається у такому форматі:

- URL сторінки;
- дата, коли сторінку було завантажено;
- http-заголовок відповіді сервера;
- тіло сторінки (html-код).

Crawler. Виділяє всі посилання на сторінці. Його завдання – визначити, куди далі має йти павук, спираючись на посилання або за заздалегідь заданим списком адрес. Краулер, проходячи по знайденим посиланням, шукає нові документи, ще відомі пошуковій системі.

Indexer. Індексатор розбирає сторінку на складові та аналізує їх, застосовуючи власні лексичні та морфологічні алгоритми. Аналізують різні елементи сторінки, такі як текст, заголовки, посилання структурні та стильові особливості, спеціальні службові html-теги і так далі.

Отже, модуль індексування дозволяє обходити за посиланнями задану кількість ресурсів, завантажувати сторінки, що зустрічаються, отримувати посилання на нові сторінки з отриманих документів та здійснювати повний аналіз цих документів.

Database. База даних – це сховище всіх даних, які пошукова система завантажує та аналізує. Іноді базу даних називають індексом пошукової системи.

Пошуковий сервер є найважливішим елементом усієї системи, оскільки від алгоритмів, покладених в основу її функціонування, безпосередньо залежить якість та швидкість пошуку.

Пошуковий сервер працює так:

1. Отриманий від користувача запит піддається морфологічному аналізу. Генерується інформаційне оточення кожного документа, що міститься в базі (яке буде згодом відображено у вигляді

відповідної текстової інформації на сторінці видачі результатів пошуку).

2. Отримані дані передаються як вхідні параметри спеціальному модулю ранжирування. Обробляються дані по всіх документах, внаслідок чого для кожного документа розраховується власний рейтинг, що характеризує релевантність запиту, введеного користувачем, та різних складових цього документа, що зберігаються в індексі пошукової системи.
3. Залежно від вибору користувача, цей рейтинг може бути скоригований додатковими умовами (наприклад, так званий «розширений пошук»).
4. Далі генерується сніпет, тобто, для кожного знайденого документа з таблиці документів вилучаються заголовок, коротка анотація, найбільш відповідне запиту та посилання на сам документ, причому знайдені слова підсвічуються.
5. Отримані результати пошуку передаються користувачеві як сторінки видачі пошукових результатів.

Основні характеристики пошукових систем

Точність – ще одна основна характеристика пошукової системи, яка визначається рівнем відповідності знайдених документів на запит користувача. Наприклад, якщо на запит «як вибрати автомобіль» знаходять 100 документів, у 50 з яких є словосполучення «як вибрати автомобіль», а в інших лише наявність цих слів («як правильно вибрати автомагнітолу та встановити її в автомобіль»), то точність пошуку буде $50/100 (=0,5)$. Чим більш точний пошук, тим швидше користувач знайде потрібні йому документи, тим менше «сміття» буде в них зустрічатися, тим рідше знайдені документи не відповідають запиту.

Актуальність – не менш важлива складова пошуку, що характеризується часом, що пройшов з моменту публікації документа в мережі Інтернет, до занесення його до індексної бази пошукової системи.

Наприклад, наступного дня після появи цікавої новини велика кількість користувачів звернулася до пошукових систем з відповідним запитом. Об'єктивно з моменту публікації новини на цю тему пройшло менше доби, проте основні документи вже були проіндексовані та доступні для пошуку завдяки існуванню у великих пошукових системах так званої «швидкої бази», яка оновлюється кілька разів на день.

Загальна модель функціонування тематичної пошукової системи

Принцип роботи розробленої тематичної пошукової системи наведено на рис.2.1.

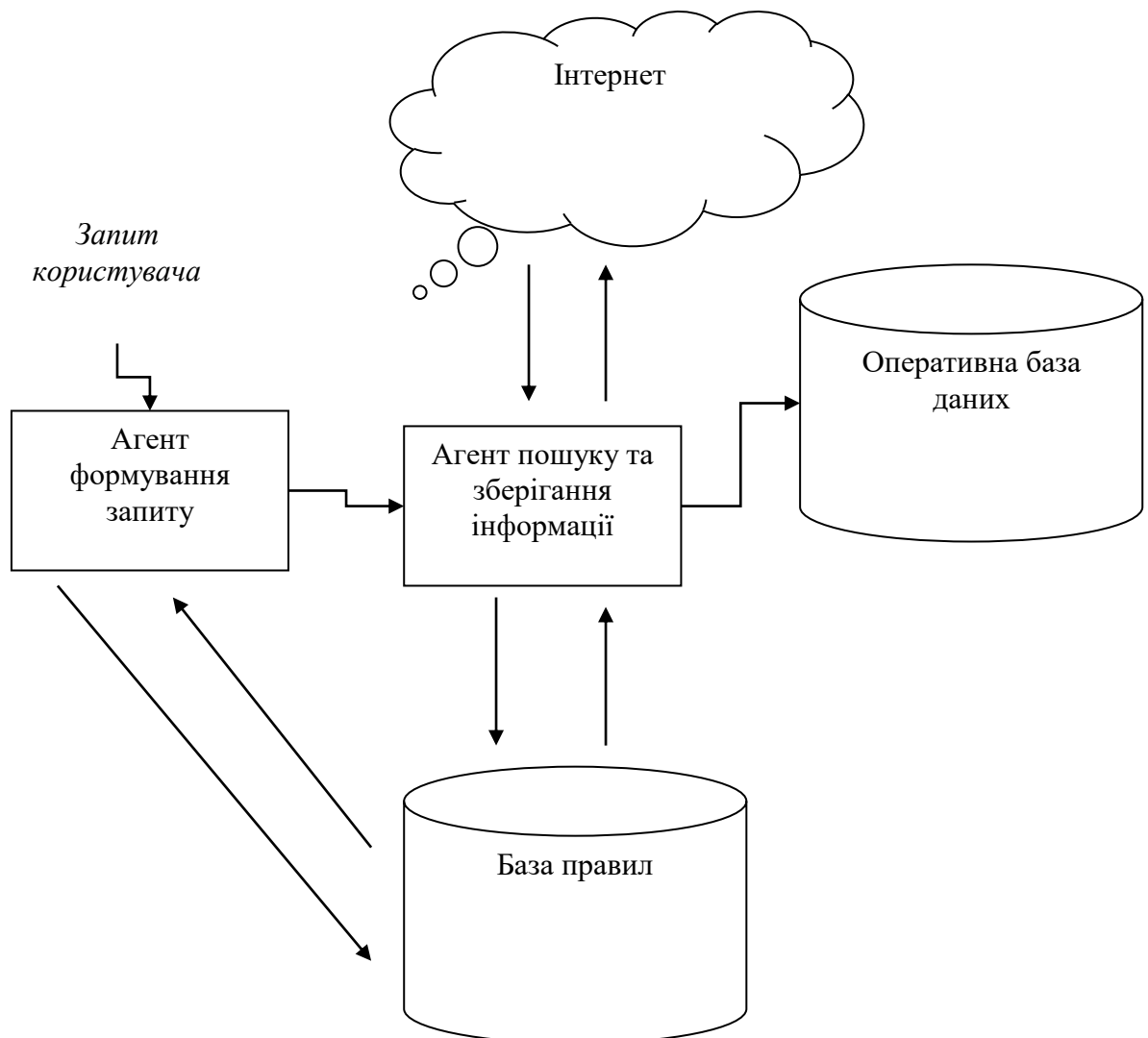


Рисунок 2.1 – Принцип роботи розробленої тематичної пошукової системи

Користувач, використовуючи можливості клієнтської частини, вводить ключове слово для пошуку інформації та вибирає зі списку тематичних сайтів ті, з яких потрібно отримати інформацію за ключовим словом.

Агент формування запиту, отримавши ключове слово та список вибраних сайтів, звертається до бази правил для отримання правил формування запитів до вибраних сайтів. Отримавши ці правила, формує запит до кожного з вибраних сайтів і передає цю інформацію агенту пошуку та зберігання інформації. У свою чергу, агент пошуку та зберігання інформації надсилає отримані запити в Інтернет та отримує html-код сторінок, які записує у свою оперативну базу даних. Потім цей агент запитує бази правил правила вилучення інформації з Шті-кодів. Обробляє інформацію, записану у своїй оперативній базі, згідно з правилами та записує обрану за правилами інформацію в оперативній базі даних системи. Далі із цією базою даних працює клієнтська частина.

База правил формується розробником системи. Саме розробник займається її правильним наповненням. Сама база правил складається з наступних полів:

- назва сайту (єдине, що відображається користувачеві в клієнтській частині);
- приклад формування запиту – тут наведено приклад запиту до відповідного сайту;
- інформація про правило поєднання ключового слова із запитом попереднього поля;
- правила вилучення інформації з html-коду, отриманої в результаті пошуку сторінки відповідного сайту.

Враховуючи динаміку руху інформації в Інтернеті, база правил повинна постійно перебувати під контролем розробника для вказівки відповідних змін властивостей сайту для внесення нових тематичних сайтів, що з'явилися в мережі. Для наповнення цієї бази можна використовувати

інтелектуальні системи, які б у кооперації зі стандартними системами пошуку відбирали сайти з конкретною тематикою.

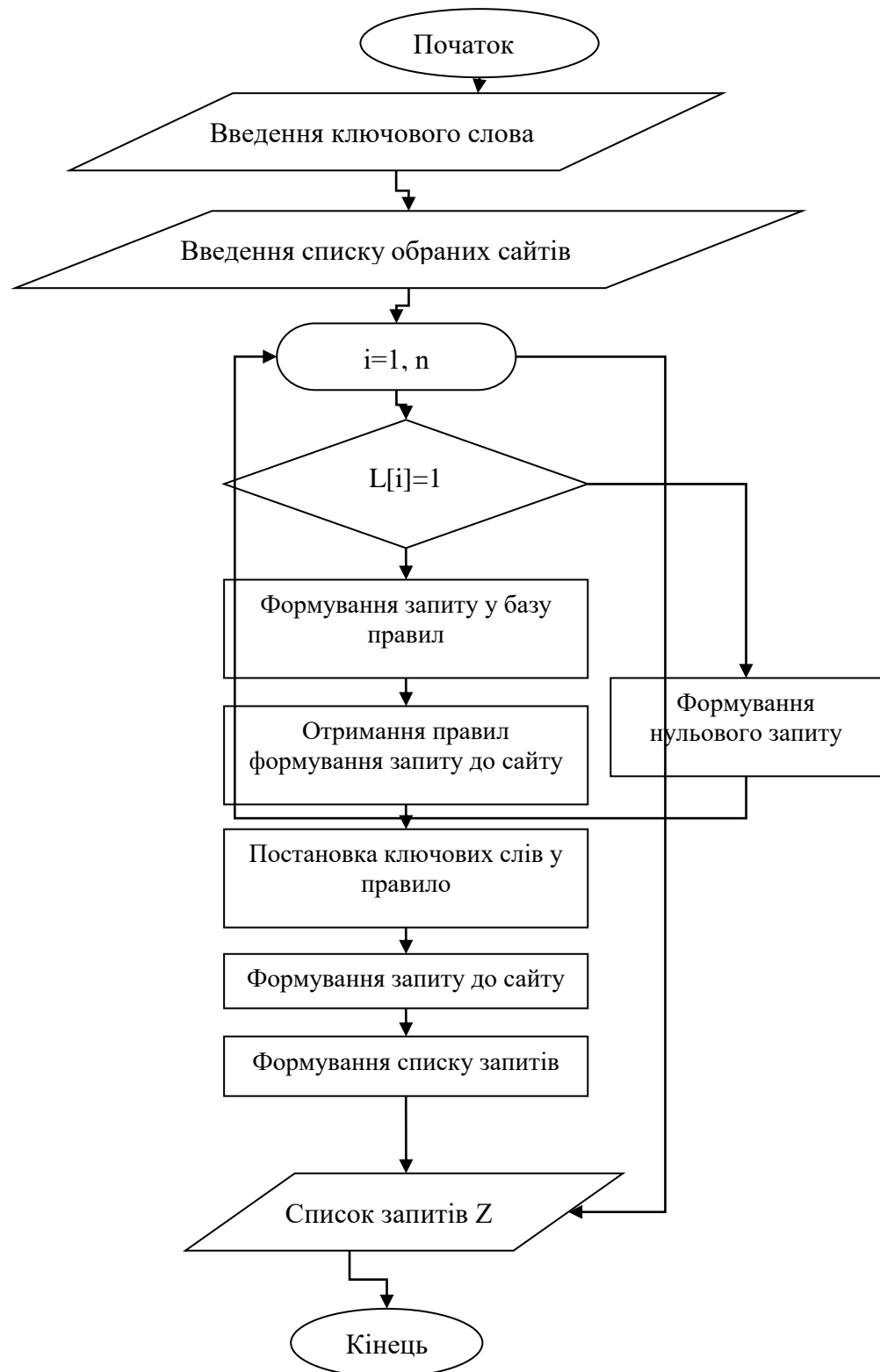


Рисунок 2.2 – Блок-схема алгоритму роботи агента формування запиту

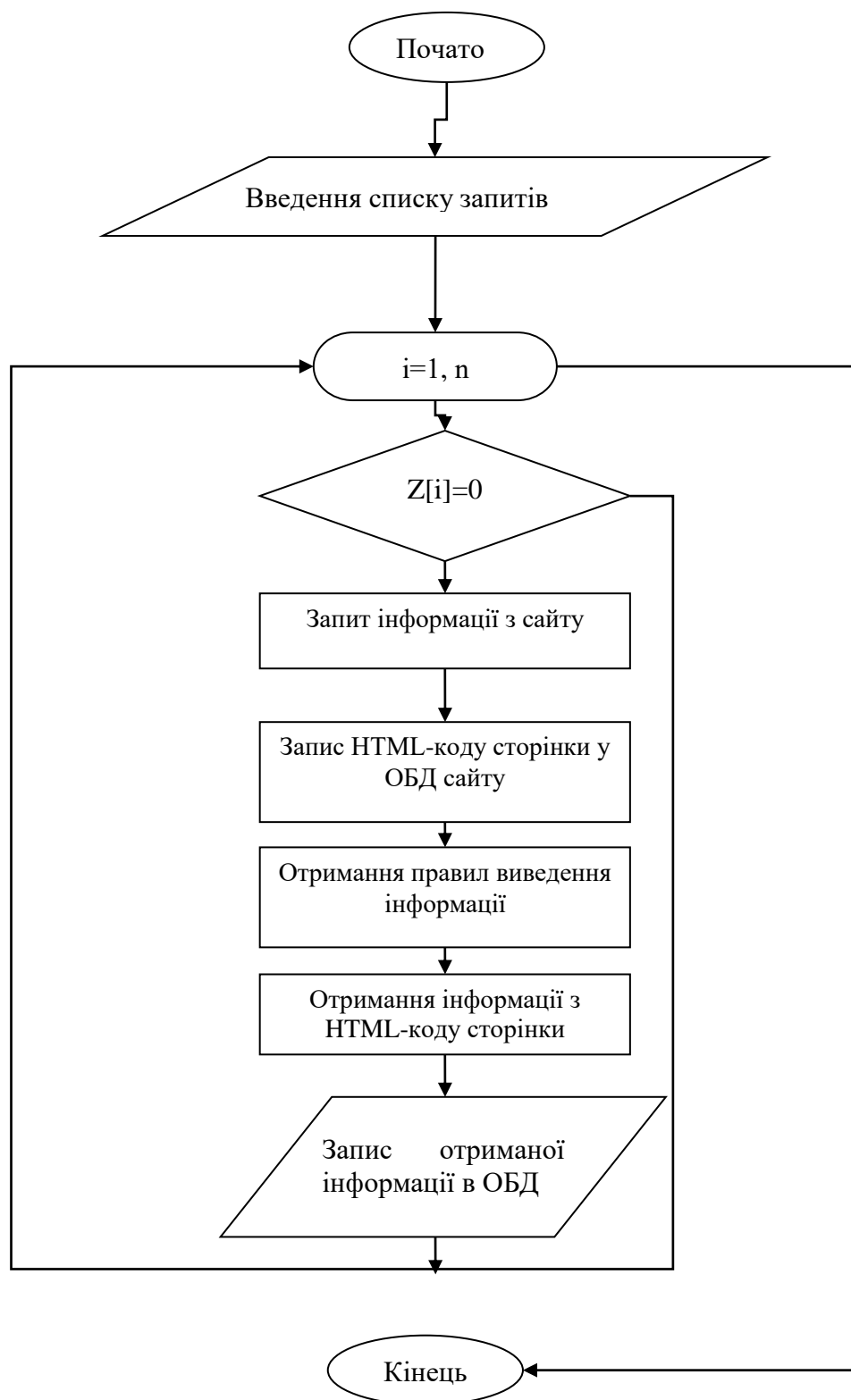


Рисунок 2.3 – Блок-схема алгоритму роботи агента пошуку та зберігання інформації

Але часто для розкручування своїх сайтів веб-майстри використовують не дуже чесні прийоми, які можуть давати неправильні результати аналізу

сайту. Саме для цього в процесі наповнення бази правил використовують людський фактор, за допомогою якого коректно відсівають «непрямі» сайти. Оперативна база даних складається з інформації, необхідної користувачеві, яка є невпорядкованою.

Агент пошуку та зберігання інформації записує в оперативну базу даних отриману з HTML-коду інформацію, яка зберігається під такими полями:

- назва сайту;
- зображення, що відповідає конкретному результату;
- текст результату пошуку;
- посилання на більш докладну інформацію.

Ця база даних формується для оперативного зберігання даних, які може переглядати, сортувати, відбирати за певними критеріями, використовуючи клієнтську частину системи.

Розглянемо алгоритм роботи кожного агента окремо. Агент формування запиту (рис. 2.2):

Отримати ключове слово від клієнтської частини.

Зчитування списку вказаних користувачем сайтів пошуку.

Перевірка кожного елемента зі списку сайтів (позначений крок 4, позначений крок 9).

Формування запиту до бази правил вибраного сайту.

Отримання правил формування запиту до конкретного сайту.

Внесення ключових слів пошуку у правило формування запиту.

Створення запиту на вибраний сайт.

Формування списку запитів для агента пошуку та зберігання інформації (перехід крок 10).

Додавання до списку запитів нульового запиту.

Повернення на крок 3, доки не закінчиться список сайтів.

Передача списку запитів агенту пошуку та зберігання інформації.

Агент пошуку та зберігання інформації (рис.2.3):

На початку своєї роботи отримує від внутрішнього агента список сформованих запитів до тематичних сайтів.

Перевірка, запит не нульовий (якщо так – крок 8).

Відсилання запиту на тематичному сайті.

Отримання коду HTML-коду сторінки, яка сформувалася як відповідь на запит.

Звернення до бази правил та отримання правила отримання інформації з коду сторінки.

Отримання необхідної інформації з html-коду сторінки (малюнки, текст, посилання).

Запис отриманої інформації до оперативної бази даних системи.

Перехід на крок 2 до завершення списку запитів.

Як видно, всі ці компоненти тісно пов'язані один з одним і працюють у взаємодії, утворюючи чіткий, досить складний механізм роботи тематичної пошукової системи, що потребує величезних витрат ресурсів.

2.2 Функціональна ефективність алгоритму

Функціональна ефективність алгоритму пошуку розглядалась на наборі даних зі списку публікацій університету за 2018-2020 роки. Цей список включає статті, які можуть бути як в електронному, так і в друкованому вигляді. Завдання полягало в тому, щоб знайти всі статті з цього списку (за авторами та назвою), які доступні в мережі Інтернет.

Всього у наборі даних 939 статей:

- 311 статей за 2018 рік;
- 324 статті за 2019 рік;
- 304 статті за 2020 рік.

З них лише для 15 статей зазначено, що вони розміщені в Інтернеті.

Здійснювався пошук наукової статті, заданої назвою та авторами, за допомогою запропонованого методу.

Якщо стаття була знайдена, вручну перевірялося, чи вона відповідає вимогам. Якщо відповідала, то вважалося, що стаття знайдено правильно, інакше – помилково.

Якщо стаття не була знайдена, то переглядалися всі завантажені в час пошуку документи. Якщо раптом серед них була виявлена стаття, то вважалося, що вона не була знайдена через помилку розпізнавання.

Потім для порівняння та оцінки результатів проводився пошук цієї статті за допомогою системи Google Scholar. Розглядалися лише перші десять результатів, видані системою.

У результаті можливі чотири результати:

Стаття знайдена правильно (клас A):

- Google Scholar теж її знайшов;
- Google Scholar її не знайшов.

Статтю знайдено помилково (клас B):

- Google Scholar знайшов потрібну статтю;
- Google Scholar не знайшов потрібну статтю.

Стаття не знайдена, але Google Scholar знайшов її (клас C):

- стаття була знайдена, але не було розпізнано, що це потрібна стаття;
- стаття не була знайдена.

Стаття не знайдена, і Google Scholar її не знайшов (клас D):

- стаття була знайдена, але не було розпізнано, що це потрібна стаття;
- стаття не була знайдена.

Для оцінки роботи методу було введено такі характеристики:

- precision (точність у класі знайдених) – частка знайдених вірно статей серед знайдених взагалі (формула (2.1));
- assigasy (точність) – частка вірних результатів серед усіх можливих результатів (формула (2.2));

- recall (повнота) – частка знайдених статей серед тих, які можна було віднайти (формула (2.3)).

$$precision = \frac{|A|}{|A| + |B|}, \quad (2.1)$$

$$accuracy = \frac{|A| + |D|}{|A| + |B| + |C| + |D|}, \quad (2.2)$$

$$recall = \frac{|A|}{|A| + |C|}. \quad (2.3)$$

Таблиця 3.1 – Оцінки, отримані в результаті експериментального дослідження

Рік	<i>precision</i>	<i>accuracy</i>	<i>recall</i>
2018	63,5%	99%	100%
2019	55%	98%	76%
2020	79%	98%	80%
Усього	66,78%	98,5%	83,5%

У ході експериментального дослідження роботи методу було виявлено такі проблеми, що призводять до помилок:

- деякі PDF-файли неможливо конвертувати у текстовий формат. Відповідно з них неможливо правильно отримати метаінформацію, щоб перевірити, чи знайдено потрібну статтю;
- зустрічаються статті тих самих авторів з дуже схожими назвами, внаслідок чого на етапі перевірки, чи не знайдена потрібна стаття, одна з таких статей може бути помилково прийнята за іншу;
- деякі статті зустрічаються виключно у збірниках, що представляються одним величезним файлом, що не дозволяє їх знайти;

- помилки та помилки у назві статті або її авторах також можуть призвести до помилок пошуку.

В цілому експериментальне дослідження показало, що запропонований метод справляється із завданням пошуку статей по заданій метайнформації і в деяких випадках навіть перевершує Google Scholar.

Однак виявилось, що дуже мало статей з набору даних розміщено в Інтернеті, що не могло не позначитися на результатах дослідження. Саме через це показник Assurasy настільки високий, а Precision, у свою чергу, досить низький.

Відмінність розробленої системи від загальноприйнятих систем пошуку полягає в тому, що вона не відображає користувача посилання на сторінки, які його цікавлять, а відображає саму інформацію з цих сторінок, оскільки тематика пошуку вже відома, а це дозволяє звужити коло до більш достовірних результатів.

Така система тематичного пошуку може використовуватися для вирішення багатьох завдань:

- виявлення оптимальної пропозиції купівлі/продажу по кількох Інтернет магазинах або аукціонах;
- наповнення методичного матеріалу різноманітних курсів дистанційного навчання з Інтернет енциклопедій та довідників;
- пошук бізнес-партнерів у різних галузях;
- порівняння пропозицій роботодавців з різних мережних дощок оголошень для молодих спеціалістів та багатьох інших завдань, які у своїй основі вимагають малої витрати часу, оптимальних рішень та наочності результатів.

Висновки до розділу 2

У рамках другого розділу розглянуто алгоритм функціонування системи веб-пошуку, яка розглядається у межах даної дипломної роботи.

Наведено блок-схеми алгоритму функціонування системи та сформовано перелік завдань які вирішує дана система.

Проведено експериментальне дослідження роботи запропонованого методу. Отримані результати показали, що метод успішно справляється з поставленим завданням. Проте в ході дослідження були виявлені проблеми, вирішення яких є подальшим розвитком методу і, швидше за все, дозволить досягти кращих результатів роботи.

3 ПРОЕКТУВАННЯ Й РОЗРОБКА ІНСТРУМЕНТАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ПОШУКУ ДОСЛІДЖЕНЬ В ГАЛУЗІ КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

3.1 Формалізація задачі

Функціональні вимоги, що додаються до системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, накладають певні обмеження і вимоги до стека технологій, що використовуються при розробці програмного забезпечення [3]. Основні вимоги, що пред'являються до системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, розробленої в рамках даної роботи:

- облік і аналіз інформації з різних джерел;
- розробка механізмів вирішення суперечностей даних;
- розробка користувальницького інтерфейсу, що дозволяє працювати з інформацією;
- розробка універсального API, що дозволяє стороннім системам запитувати інформацію.

Діаграма використання наведена на рис. 3.1.

На даній діаграмі зображено відношення між векторами та прецедентами в системі. На цій діаграмі виділені два вектори : користувач та адміністратор (розробник системи). Виділені прецеденти дозволяють специфікувати функціональну поведінку розроблюваної системи та отримати відповідь на запитання, що має робити система, але не визначають реалізацію цієї поведінки системи – не торкаються питань, яким чином відповідні функції реалізуються.

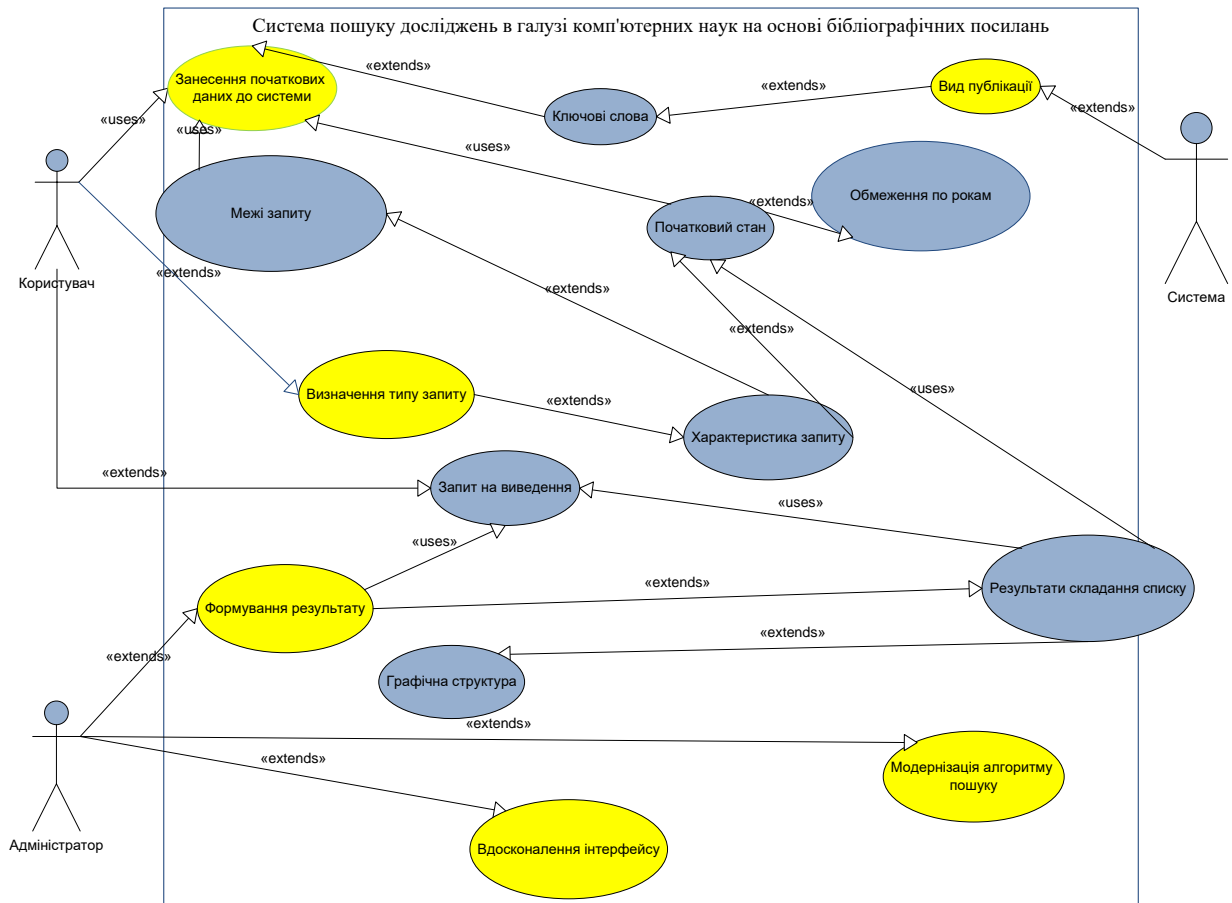


Рисунок 3.1 – Діаграма використання системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

Розглянемо основні сценарії, доступні співробітникам в рамках розробленої системи. Вони наведені в таблиці 3.1 і таблиці 3.2.

Таблиця 3.1 – Сценарій використання комп'ютер оператора – система

Діючі особи	Комп'ютер оператора, Система
Мета	Додати ключові слова
Передумова	Для оператора відсутні «незакриті» записи в базі даних (всі записи мають час початку і закінчення)

Успішний сценарій:

1. Оператор включає комп'ютер.
2. Програма клієнт відправляє інформацію про наявність наукових публікацій на сервер.

3. Сервер робить запис в базу даних про наявність наукових публікацій.
4. Сервер посилає відповідь про успішний запис інформації в базу даних програмі клієнту.

Результат На екрані з'являється список наукових публікацій

Таблиця 3.2 – Сценарій використання оператор – система

Діючі особи	Оператор, Система
Цілі	Переглянути дані журналу наукових публікацій
Успішний сценарій:	
1. Оператор через web-інтерфейс робить запит до системи 2. Система, по запитуваним параметрам робить запит до бази. 3. Система видає оператору відповідь з інформацією про наявність наукових публікацій.	
Результат	Оператор бачить деталізацію наукових публікацій.
Розширення:	
* А	Оператор хоче внести зміни в обмеження пошуку наукових публікацій. Оператор вносить інформацію в запит через web-інтерфейс. Результат: інформація збережена.

Основна функція адміністратора системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань – підтримка актуальної інформації в пошуковому запиті. Він повинен мати інтерфейс для здійснення CRUD-операцій [5]. Сценарій для даного варіанту наведено в таблиці 3.3.

Таблиця 3.3 – Сценарій використання адміністратор – система

Діючі особи	Адміністратор, Система
Мета	Редагування сутностей бази даних
Передумова	Сервер бази даних і сервер додатка запущені і мають необхідне з'єднання
Успішний сценарій:	
1.Адміністратор через web-інтерфейс заходить на сторінку редагування даних. 2. Робиться запит на редагування даних до сервера. 3. Сервер обробляє запит, роблячи необхідні зміни в базі даних. 4. Сервер повертає відповідь з актуальною інформацією з БД	
Результат	У базі даних зберігаються зміни, зроблені адміністратором.

Діаграма послідовностей системи (sequence diagram)

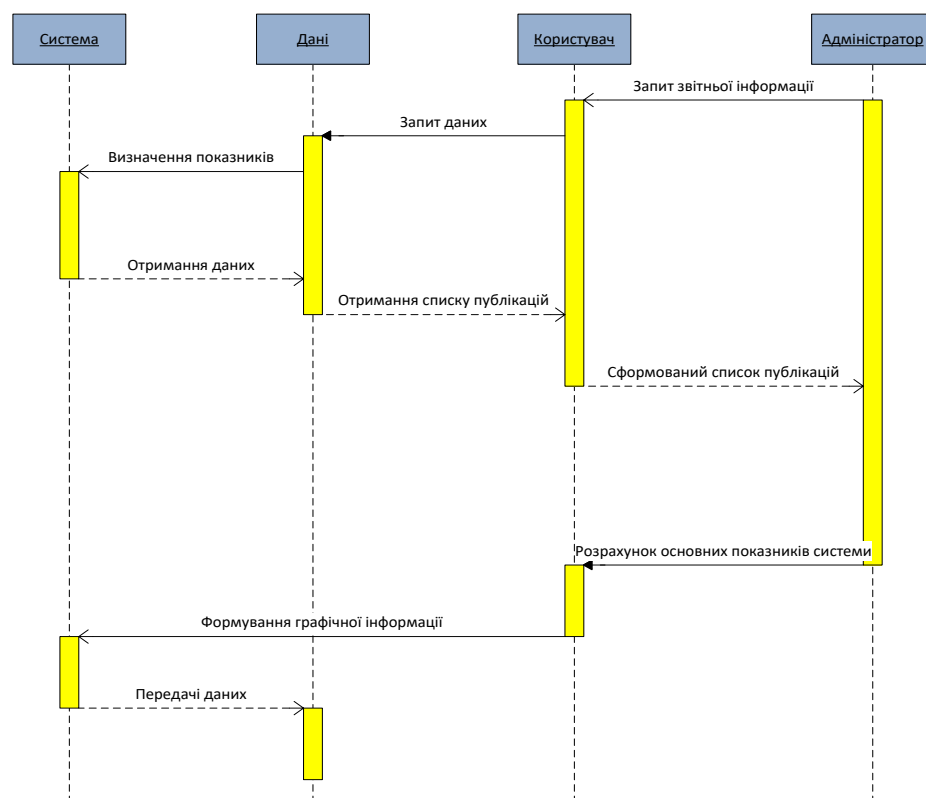


Рисунок 3.2 – Діаграма послідовностей

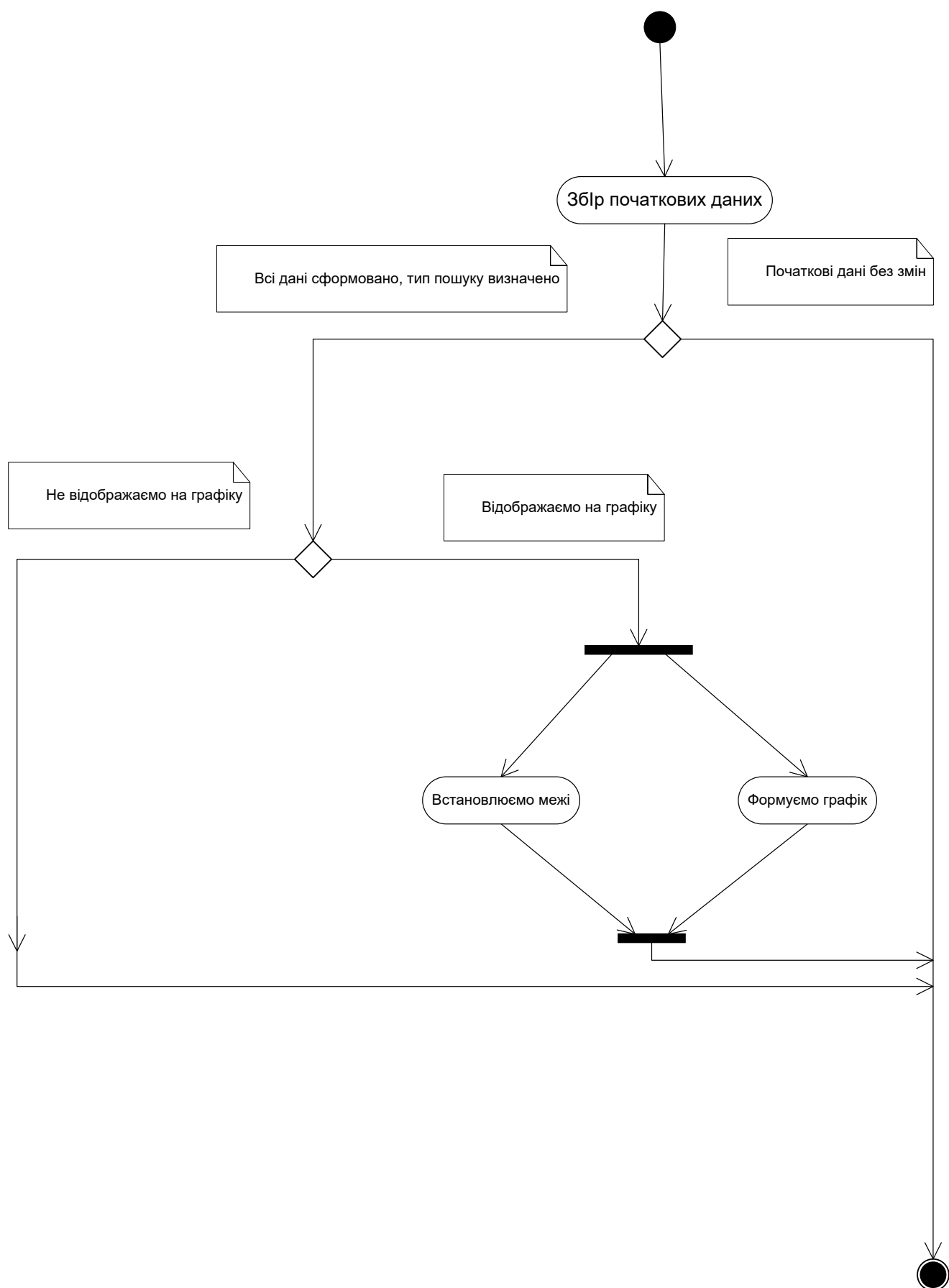


Рисунок 3.3 – Діаграма активності

На рисунку 3.4 представлений макет головної сторінки системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

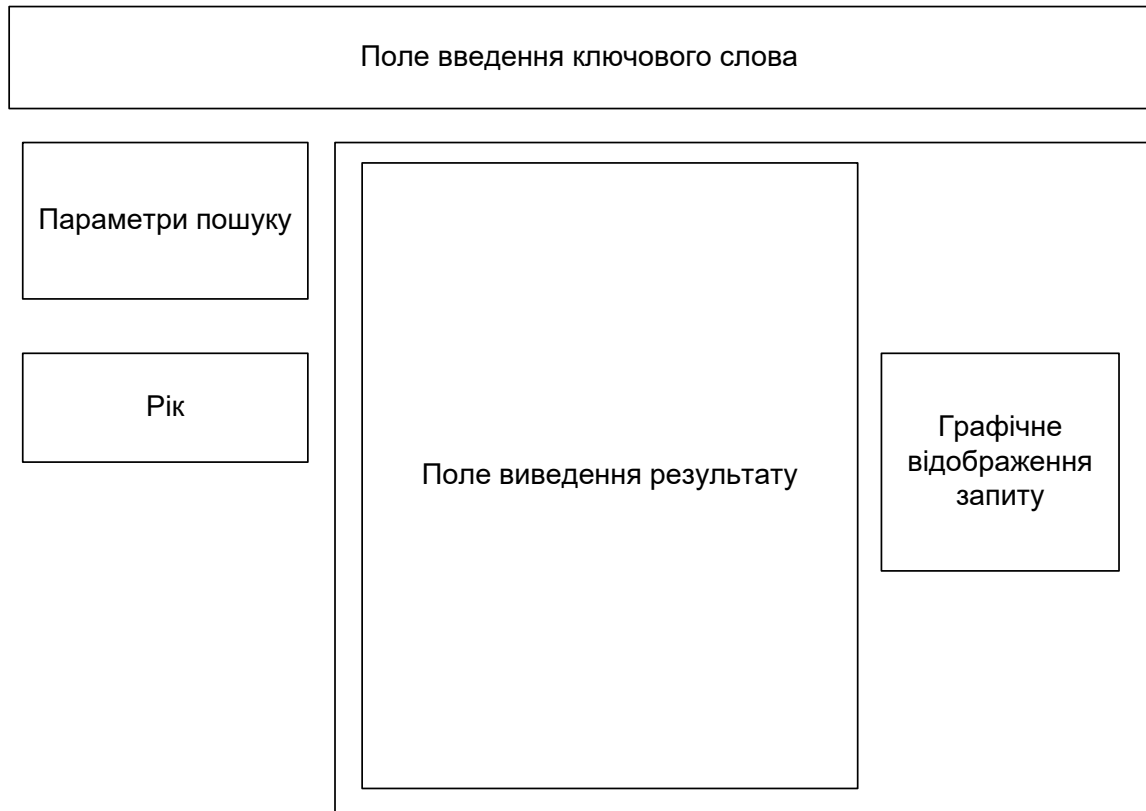


Рисунок 3.4 – Макет головної сторінки системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

Діаграма компонент – статична структурна діаграма, показує розбиття програмної системи на структурні компоненти та зв'язки (залежності) між компонентами. У якості фізичних компонентів можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети [16].

Діаграма компонент додатку представлена на рисунку 3.5.

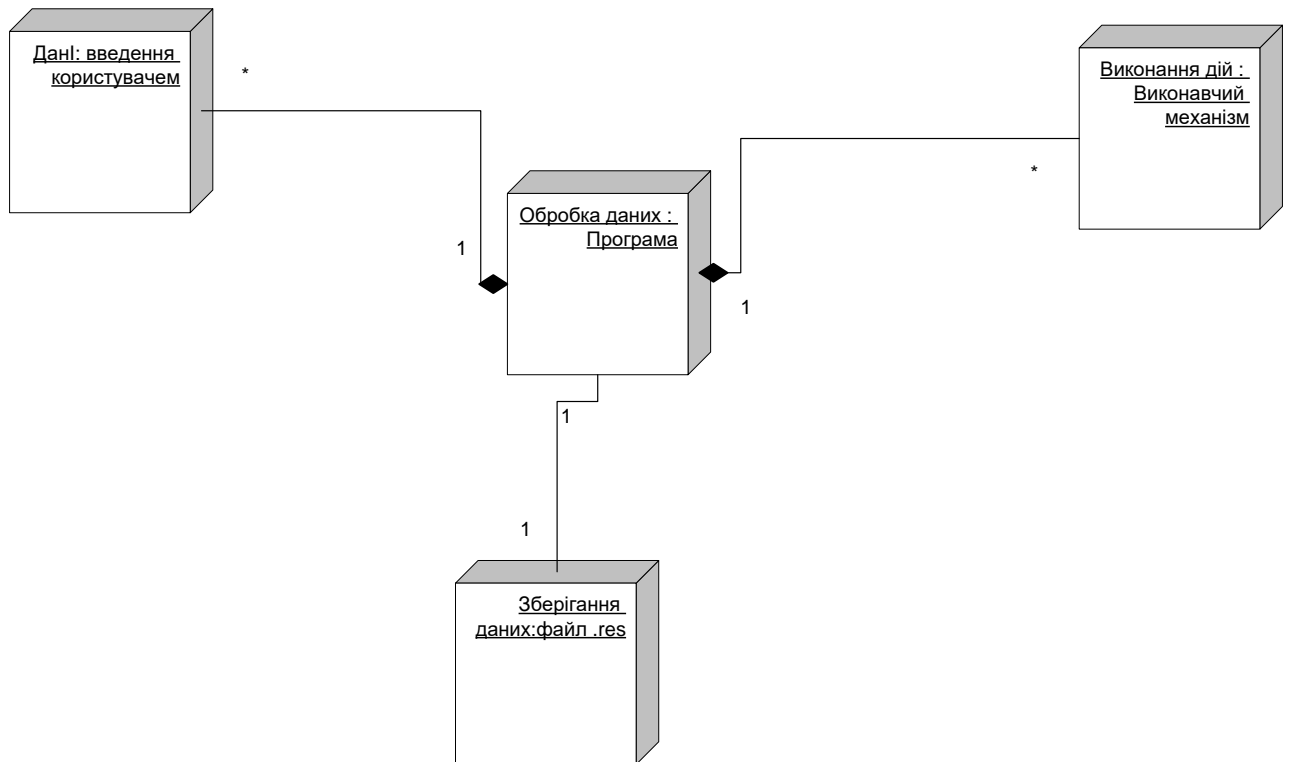


Рисунок 3.5 – Діаграма компонент

На даній діаграмі показано склад фізичних ресурсів системи (що позначаються як вузли системи) та відношень між ними.

Діаграма розгортання містить графічні зображення процесорів, пристроїв, процесів і зв'язків між ними. На відміну від діаграм логічного представлення, діаграма розгортання є єдиною для системи в цілому, оскільки повинна цілком відбивати особливості її реалізації. Ця діаграма, по суті, завершує процес ООАП для конкретної програмної системи та її розробка, як правило, є останнім етапом специфікації моделі.

Отже, перерахуємо цілі, переслідувані при розробці діаграми розгортання:

- визначити розподіл компонентів системи по її фізичним вузлам;
- показати фізичні зв'язки між усіма вузлами реалізації системи на етапі її виконання;
- виявити вузькі місця системи і реконфігурувати її топологію для досягнення необхідної продуктивності.

Для забезпечення цих вимог діаграма розгортання розробляється спільно системними аналітиками, мережевими інженерами і системотехніками.

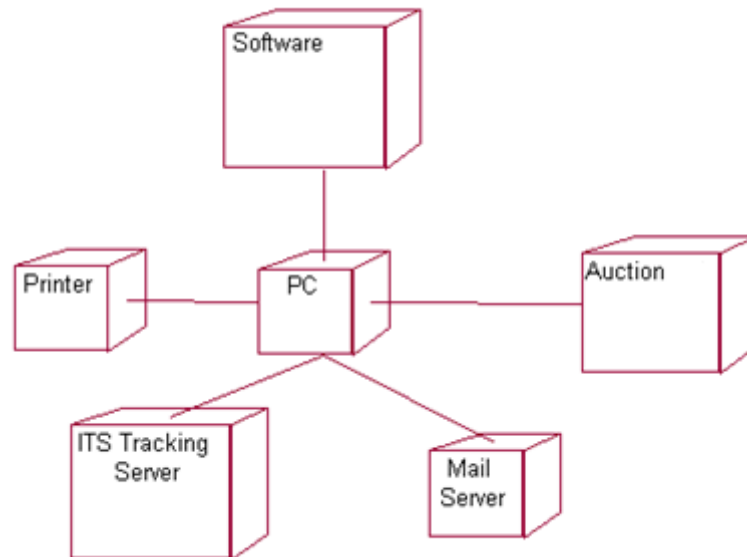


Рисунок 3.6 – Діаграма розгортання на одній машині

3.2 Базова архітектура системи

«Клієнт – сервер» (англ. Client-server) – обчислювальна або мережева архітектура, в якій завдання або мережеве навантаження розподілені між постачальниками послуг, званими серверами, і замовниками послуг, званими клієнтами. Фактично клієнт і сервер – це програмне забезпечення.

Архітектура програми часто створюється для ілюстрації загальних властивостей програмного забезпечення (наприклад, код програми та платформа) і обладнання (наприклад, клієнт, сервер і мережеві пристрої). У літературі виділяють ряд базових шаблонів.

Архітектура додатків, як правило, проектується в термінах архітектури «клієнт-сервер», при якій один або кілька клієнтських пристроїв подають запит на інформацію з сервера. Сервер зазвичай володіє необхідною інформацією (рис. 3.7).

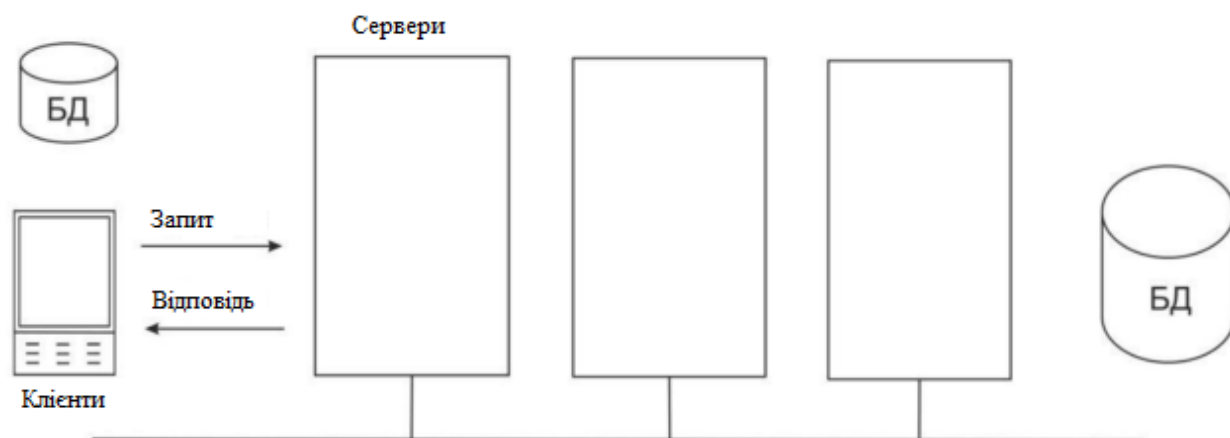


Рисунок 3.7 – Клієнт-серверна архітектура [26]



Рисунок 3.8 – Дволанкова архітектура

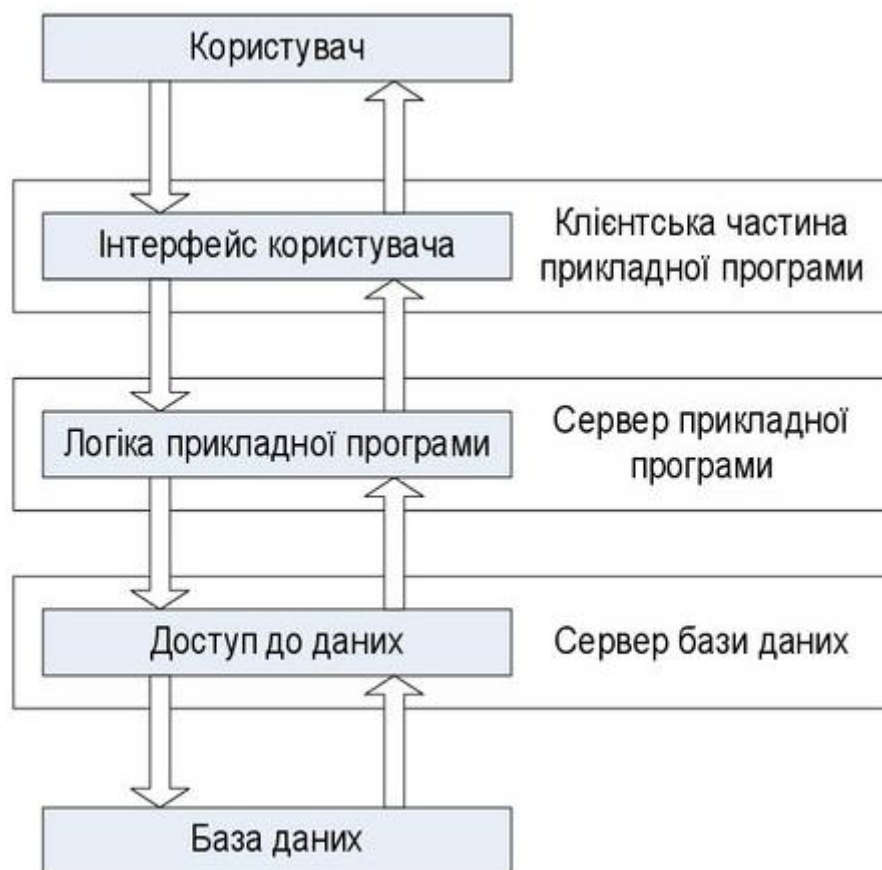


Рисунок 3.9 – Триланкова архітектура

Розглянемо такі поняття клієнт-серверної архітектури: шари, рівні і зв'язок між шарами і рівнями.

Функціональність коду по всьому додатку необов'язково рівномірна. Деякі розділи коду програми краще підходять для обробки користувацького інтерфейсу, в той час як інші розділи розроблені для управління бізнес-логікою або з'єднання з базою даних або серверних систем.

Розшарування описує поділ робіт всередині коду програми на одній машині. Найчастіше шари – це не більше ніж програмні модулі, які розміщені в різних папках або каталогах на стороні клієнта або сервера.

З боку клієнта зазвичай є від нуля до трьох шарів в коді програми. З боку сервера – від одного до трьох шарів коду програми. Частково це важливо для гарного проектування програмного забезпечення, яке гарантує повторне використання коду, частково це важливо для безпеки, частково з міркувань зручності.

Клієнт з нульовою кількістю шарів коду, по суті, не має спеціального коду програми. Цей тип клієнта зазвичай згадується як тонкий клієнт і можливий в архітектурі клієнт-сервер, при якому сервер містить весь призначений для користувача код програми. Клієнт з одним до трьох шарів коду програми зазвичай називається товстим клієнтом.

Сервер, також може містити від одного до трьох шарів спеціального коду програми. Проте, за визначенням, не може бути нульової кількості шарів коду на сервері.

Шар, код якого найтісніше взаємодіє з користувачем, часто згадується в літературі як шар уявлення. Другий шар зазвичай обробляє бізнес-логіку коду. Третій шар називають рівнем доступу до даних (зв'язок з базою даних або з джерелом даних).

Цілком можлива наявність більш ніж трьох рівнів на стороні клієнта або сервера, але занадто багато шарів може привести до труднощів при управлінні.

3.3 Внутрішнє проектування

Відповідно до розглянутих раніше концептуальних вимог до системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, найбільш підходящою архітектурою для розробки програми є архітектура клієнт-сервер. Використання даного архітектурного підходу дасть ряд переваг [6]:

- Загальна серверна логіка при обробці запитів
- Загальна база даних, що надається в якості ресурсу,
- Незалежність клієнтів один від одного

Розглянемо загальну схему взаємодії в рамках розробленої системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань представленої на рисунку 3.10.

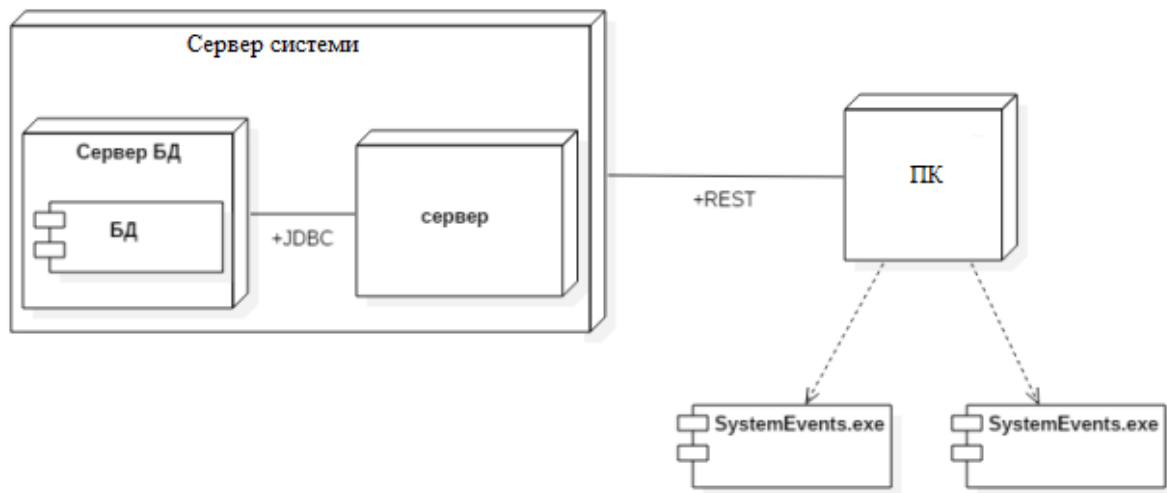


Рисунок 3.10 – Архітектура системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

На рисунку 3.10 наведені основні структурні частини системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань:

- сервер бази даних, на якому безпосередньо знаходиться база даних і який забезпечує запити до неї з боку сервера додатку і відповіді на них.
- сервер програми – ядро розробленої системи. Містить основну бізнес логіку додатка, а також пов'язує всі інші частини інфраструктури розробленої інформаційної системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.
- комп'ютери-клієнти – робочі комп'ютери операторів, на кожен з яких встановлений свій екземпляр програми-клієнта, що передає інформацію на сервер.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань в термінології класів об'єктно-орієнтованого програмування. Діаграма класів може відбивати, зокрема, різні взаємозв'язки між окремими сутностями предметної області, такими як

об'єкти і підсистеми, а також описує їхню внутрішню структуру і типи відносин. На даній діаграмі не вказується інформація про тимчасові аспекти функціонування системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. З цієї точки зору діаграма класів є подальшим розвитком концептуальної моделі проектованої системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

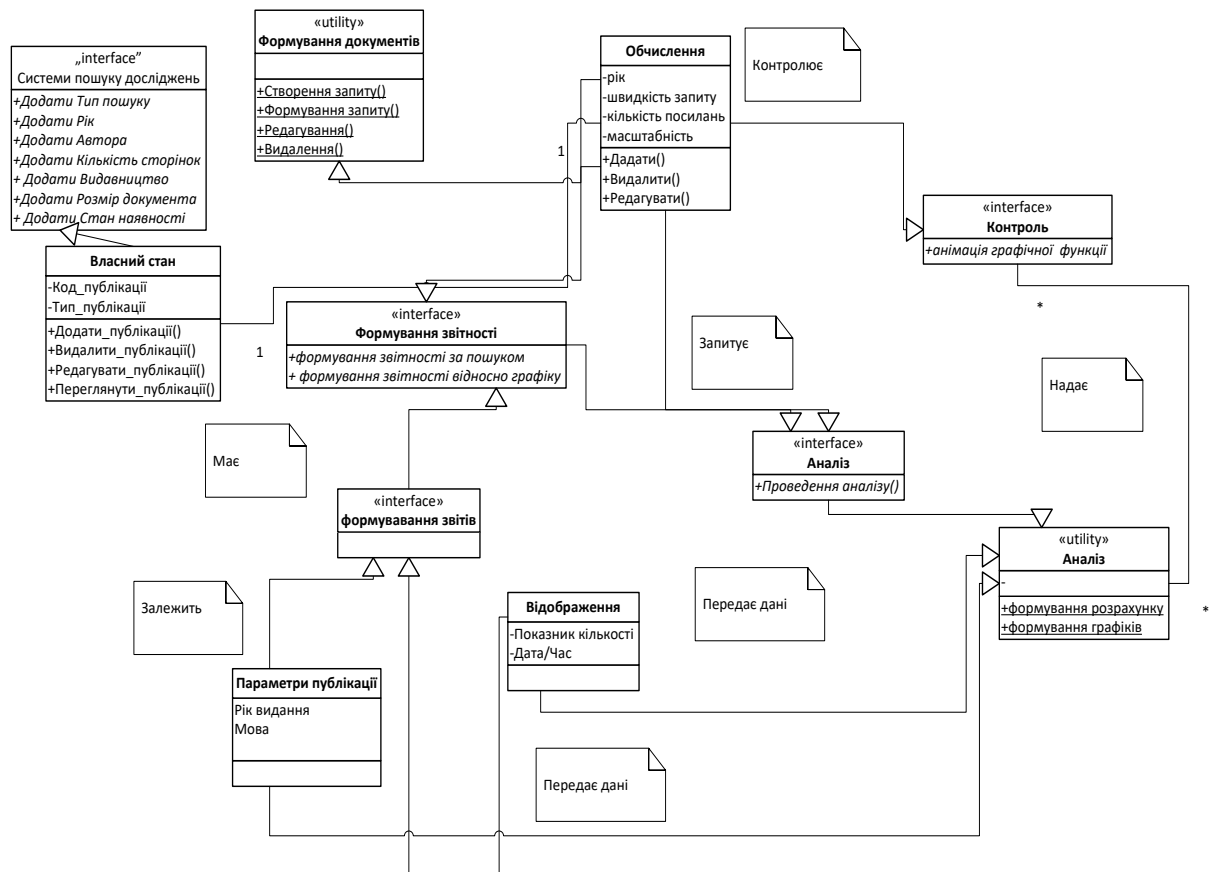


Рисунок 3.11 – Діаграма класів

3.4 Розробка інтерфейсу користувача

Інтерфейс користувача, або UI (User Interface) – це зовнішній вигляд продукту, спосіб спілкування між користувачем і програмою.

Інтерфейс має велике значення для довільної програмної системи і є обов'язковою її складовою, спрямованою насамперед на кінцевого користувача. Саме за інтерфейсом користувач судить про програму в цілому;

до того ж, часто висновок про використання програмного забезпечення користувач сприймає по тому, наскільки йому комфортний і зрозумілий інтерфейс. Разом з тим, трудомісткість проектування та розробки інтерфейсу досить чимала.

Не буває єдиної загальновизнаної класифікації засобів розробки інтерфейсу користувача. Так, у [1] програмне забезпечення для розробки інтерфейсу поділяється на дві ключові категорії – інструмент інтерфейсу (toolkits) та високорівневі засоби розробки (higher-level development tools). Інструмент користувацького інтерфейсу найчастіше містить бібліотеку елементів (меню, кнопки, смуги прокручування та інших.) і необхідний застосування програмістами.

Високорівневі засоби конструювання інтерфейсу можуть бути використані непрограмістами і оснащені стилем, який дозволяє специфікувати функції введення-виводу, а також визначати, використовуючи техніку прямого маніпулювання, інтерфейсні елементи. До подібних засобів відносять будівельники діалогу (interface builders) і СУПІ – системи управління інтерфейсом користувача (User Interface Management Systems – UIMS). Крім СУПІ, деякі автори користуються такими визначеннями, як User Interface Development Systems (UIDS) – системи розробки інтерфейсу користувача, User Interface Design Environment (UIDE) – область розробки інтерфейсу користувача та ін.

Можна виділити кілька основних методів специфікації інтерфейсу [2].

1. Мовний, коли використовуються особливі мови завдання синтаксису інтерфейсу (декларативні, об'єктно-орієнтовані, мови подій та ін.).

2. Графічна специфікація пов'язана з описом інтерфейсу засобами візуального програмування, програмуванням демонстрацій та за зразками. Такий спосіб підтримує тонкий клас інтерфейсів.

3. Специфікація інтерфейсу, що базується на об'єктно-орієнтованому підході, пов'язана з принципом, що називається безпосереднє маніпулювання. Ключова його властивість – взаємодія користувача з

окремими об'єктами, а чи не з повною системою як неподільним цілим. Типовими складовими, що застосовуються для дій з об'єктами та керуючими функціями, є обробники, меню, зони діалогу, кнопки різного виду.

4. Специфікація інтерфейсу зі специфікації практичного завдання. Тут інтерфейс формується автоматично за специфікацією семантики практичного завдання. Все ж таки складність відтворення інтерфейсу ускладнює можливості швидкої появи систем, що реалізують певний підхід.

Для забезпечення ефективної роботи користувача від розробника інтерфейсу потрібно не порушувати рівновагу між наведеними нижче факторами протягом усього життєвого циклу розробки та конструювання програми [3]. Це досягається послідовним і кропітким опрацюванням елементів інтерактивної взаємодії на будь-якому з етапів розробки інтерфейсу (рисунок), що включають [4- 5]:

Проектування

1. Високофункціональні вимоги: формулювання мети розробки та початкових вимог.
2. Аналіз користувачів: визначення потреб користувачів, створення сценаріїв, оцінка відповідності сценаріїв очікуванням користувачів.
3. Концептуальне проектування: імітація процесу, для котрого розробляється додаток.
4. Логічне проектування: встановлення інформаційних потоків у додатку.
5. Фізичне проектування: підбір платформи, на якій буде виконано проект та засоби розробки.

Реалізація

1. Прототипування: розробка паперових та/або діалогових макетів екранних форм.
2. Конструювання: створення програми з урахуванням можливості виправлення його дизайну.

Тестування

1. Юзабіліті-тестування: налагодження програми різними користувачами, в т.ч. та користувачами з обмеженими можливостями (Accessibility testing).

Як і розробка програми в цілому, так і генерування інтерфейсу користувача для нього – процес ітераційний. Малоімовірно, що подібні етапи, як прототипування, конструювання та тестування інтерфейсу, можуть бути закінчені за один прохід. Тому, якщо внаслідок випробування виявлено недоробки, то вони, якщо таке можливе, усуваються шляхом вторинного конструювання або розробляється інший прототип інтерфейсу.

Головна сторінка програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань умовно розділена на два поля. Перше поле являє собою поле введення запиту, а друга частина це безпосередньо поле виведення результату запиту.

Головна сторінка програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань наведена на рисунку 3.12.

Не защищено | scholar-master:8080

Ключевые слова: [Начать поиск](#)

Год Сортировать по Страница

Данный сайт позволяет осуществить анализ динамики развития исследований в области компьютерных наук на основе библиографических ссылок.

© 2021 [Дипломная работа](#)

Рисунок 3.12 – Головна сторінка програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

3.5 Тестування та налагодження програми

Для проведення тестування розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань обрано локальний сервер OpenServer.

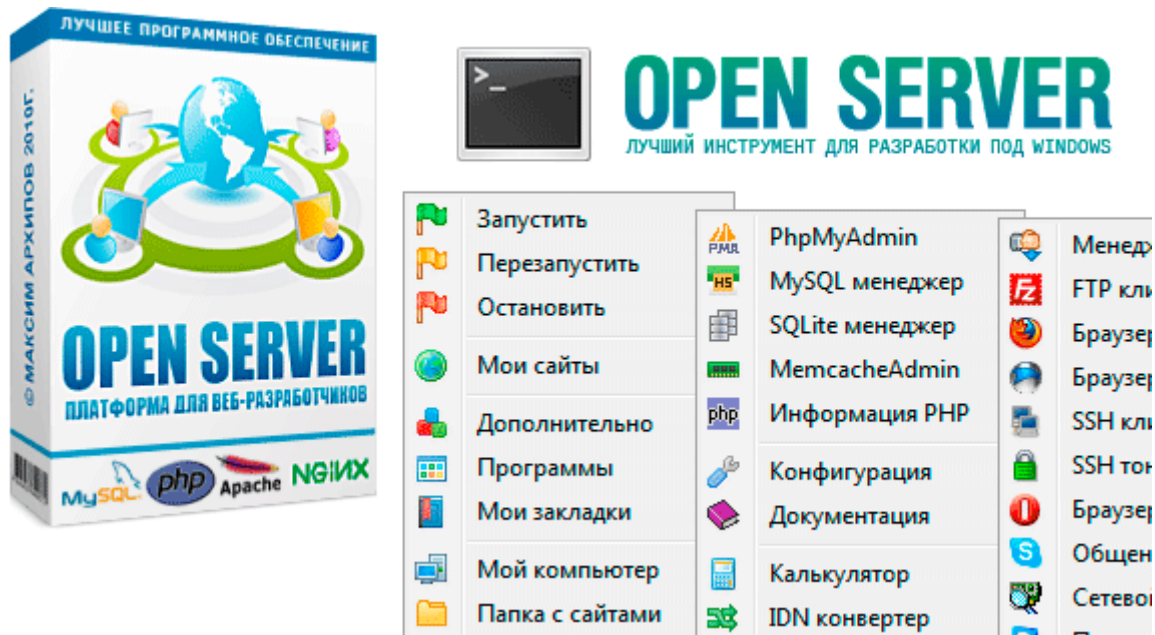


Рисунок 3.13 – Open server

Open Server – це портативний локальний WAMP/WNMP сервер, що має багатофункціональну керуючу програму і великий вибір компонентів, що підключаються.

Для налагодження скриптів у різному оточенні Open Server пропонує на вибір відразу два види HTTP серверів, різні версії PHP і СУБД модулів, а також можливість швидкого перемикання між ними.

HTTP модулі: Apache 2.2.21 та Nginx 1.0.11;

СУБД модулі: MySQL 5.1.61, MySQL 5.5.20 та PostgreSQL 9.1.1;

PHP модулі: PHP 5.2.17 (IMagick 2.2.1, Zend Optimizer 3.3.3, IonCube Loader 4.0.7, Memcache 2.2.4) та PHP 5.3.9 (IMagick 2.3.0, Xdebug 2.1.3, IonCube 10, Memcache 2.2.6);

Відмінний набір інструментів: HeidiSQL, Adminer, PHPMyAdmin, PHPPgAdmin, PgAdmin

До складу пакету також включені такі смакоти як: Perl, FTP сервер, Sendmail, Memcached сервер.

Open Server – це єдиний відомий проект, до якого включено Nginx. Причому тут реалізовано зручне підключення правил реврайту через файли .nхaccess в корені домену, а PHP працює в режимі True FastCGI (невбивається).

Усі компоненти взяті з офіційних репозиторіїв та завжди оновлюються до актуальних версій з кожним оновленням пакета.

Open Server – це цілком портативний сервер. Жодних системних сервісів, купа сміття в реєстрі і system32.

У разі відсутності на комп'ютері потрібних системних компонентів Open Server встановить їх сам, достатньо вибрати меню [Інструменти – Перший запуск] якщо сервер запускається на комп'ютері вперше.

За допомогою Open Server можна запустити/зупинити сервер або відкрити потрібний домен.

Особливості Open Server:

- докладний перегляд логів всіх компонентів в реальному часі;
- вибір HTTP, СУБД та PHP модулів у будь-якому поєднанні;
- підтримка SSL і кириличних доменів з коробки;
- підтримка аліасів або доменних покажчиків, а також зручна форма їх налаштування (привіт любителям мультисайтингу в Drupal!);
- створення локального піддомену без втрати видимості основного домену в Інтернеті;
- доступ до доменів (в один клік) та швидкий доступ до шаблонів конфігурації модулів;

- мультимовний інтерфейс (Російський, Український, Білоруський, Англійський).

Програма постійно вдосконалюється, всі адекватні прохання з боку користувачів Open Server детально вивчаються та більшість із них реалізується.

Папка з розробленим програмним додатком для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань лежить у кореневій папці сервера (рис. 3.14).

Имя	Дата изменения	Тип
hospitalCateringSys-master	07.05.2021 10:47	Папка с файлами
id1986	07.05.2018 17:17	Папка с файлами
install	30.01.2017 15:53	Папка с файлами
localhost	21.10.2016 0:14	Папка с файлами
magento2	14.04.2021 16:39	Папка с файлами
PHP5Bootstrap4MySQLClassExhibitionTut...	04.04.2018 10:43	Папка с файлами
php-cafe-master	09.05.2020 17:20	Папка с файлами
restaurant	10.05.2020 9:06	Папка с файлами
Scholar-master	26.11.2021 16:25	Папка с файлами
security	10.11.2019 11:07	Папка с файлами
server	02.06.2018 18:35	Папка с файлами
smart-hospital-master	07.05.2021 11:59	Папка с файлами
Software Desarrollado	07.05.2021 11:58	Папка с файлами
stcms	17.06.2019 15:20	Папка с файлами
StockDeck-master	26.11.2021 14:24	Папка с файлами
www	26.11.2021 11:53	Папка с файлами
www.Art-exhibition.ua	10.05.2020 19:45	Папка с файлами
www.banking.ua	18.03.2019 18:58	Папка с файлами
www.landshaft.ua	16.09.2020 9:22	Папка с файлами
www.RestaurantSite.ua	09.05.2020 15:00	Папка с файлами
yummy-shop-admin-master	07.05.2021 13:35	Папка с файлами

Рисунок 3.14 – Коренева папка серверу

Саме з відси і відбувається запуск розробленого програмного продукту. Склад папки становить виконуючі файли (рис. 3.15).

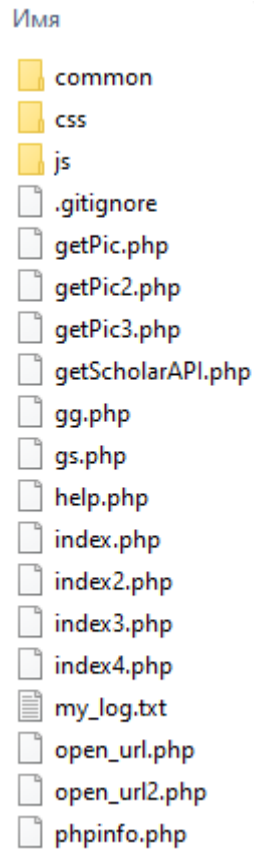



Рисунок 3.15 – Склад виконуючих файлів

Після запуску локального серверу, , переходимо у вкладку з сайтами (рис. 3.16) та натискаємо на назві нашого сайту (рис. 3.17) на екрані браузера з'явиться головна сторінка (рис. 3.18).

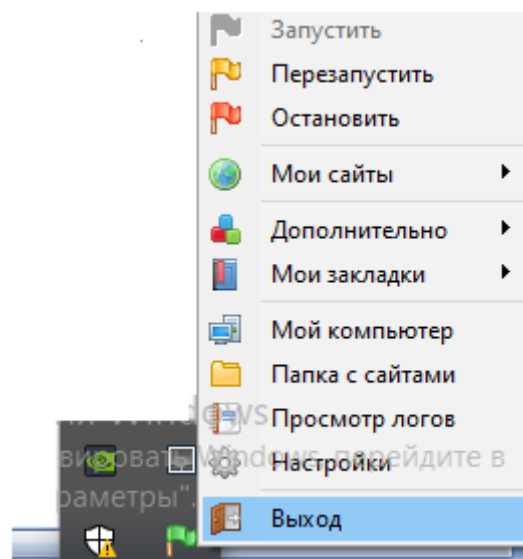


Рисунок 3.16 – Вкладення локального серверу

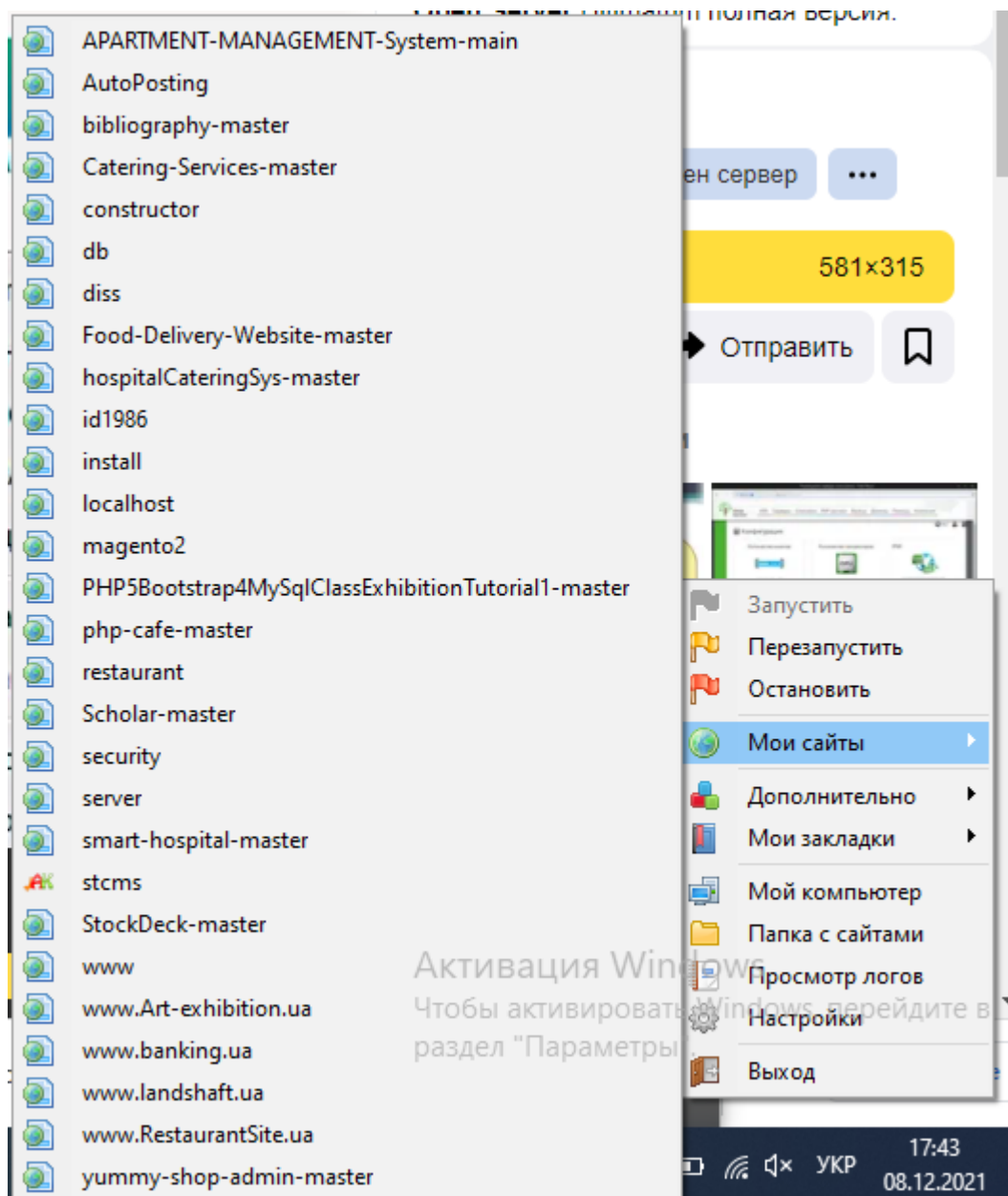


Рисунок 3.17 – Перехід по назві сайту

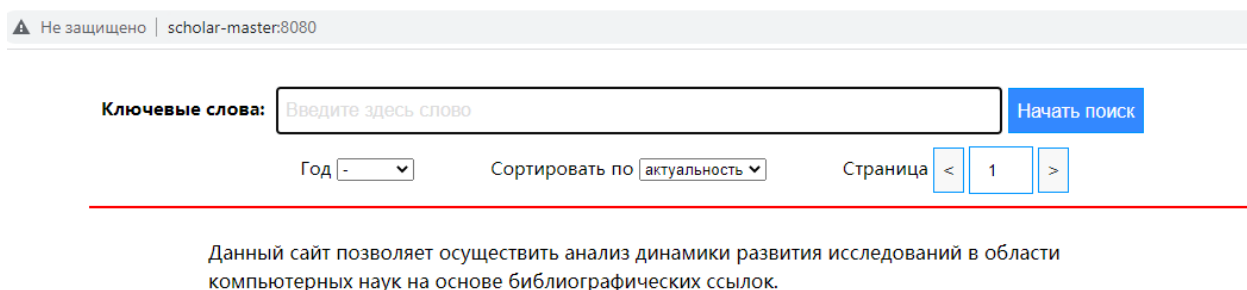
© 2021 [Дипломная работа](#)

Рисунок 3.18 – Головна сторінка

Тепер протестуємо програмний додаток для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань ввівши у поле запити ключове слово, наприклад «База даних» (рис. 3.19). Натискаємо на активній кнопці «Почати пошук» програма запропонує нам почекати поки формується загальний список публікацій (рис. 3.20) та як результат запити та очікування отримуємо результат пошуку (рис. 3.21).

Дипломная работа'."/>

Рисунок 3.19 – Запит на пошук публікацій

Дипломная работа'."/>

Рисунок 3.20 – Очікування результату

Ключевые слова: [Начать поиск](#)

Год Сортировать по Страница < >

<https://scholar.google.com/scholar?ie=UTF-8&q=%D0%91%D0%B0%D0%B7%D0%B0%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85>

Примерное количество результатов: 193 000 (0,06 сек.) Мій

диапазон

дповідн.

з тою

патенти

Реляційні бази даних як сучасний стандарт накоплення інформації в комп'ютерній системі бухгалтерського обліку [PDF] wunu.edu.ua

[ОВ Адамик](#) , [КБ Адамик](#) - 2018 - [dspace.wunu.edu.ua](#)

Реляційні бази є стандартом при побудові інформаційних систем підприємства і комп'ютерних систем бухгалтерського обліку, зокрема. Цьому сприяли такі їх переваги: 1) відсутність дублювання інформації; 2) при зміні значення поля, достатньо змінити її...

☆ Зберегти 99 Послатися Цитовано в 13 джерелах Пов'язані статті Кількість версій: 6 99

База даних як об'єкт інформаційних правовідносин [PDF] univd.edu.ua

МЮ фіменко - 2015 - [dspace.univd.edu.ua](#)

На підставі аналізу дефініцій « база даних », «об'єкти інформаційних правовідносин» доводиться, для бази даних може розглядатися як об'єкт інформаційних правовідносин. Наголошено, надає доступ к базам даних на межсайтових інформаційних правовідносин потребу...

☆ Зберегти 99 Послатися Цитовано в 5 джерелах Пов'язані статті Кількість версій: 2 99

Бази і сховища даних –інформаційний фундамент бухгалтерського обліку та аналізу [PDF] wunu.edu.ua

Рисунок 3.21 – Результат роботи розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

Програмний додатком передбачено можливість виведення результату за роком, для цього необхідно із випадającego списку (рис. 3.22) обрати необхідний рік, наприклад обираємо 2021 рік (рис. 3.23) програма просить знову почекати поки відбувається вибірка наукових публікацій за вказаним роком, результат отримуємо по року, який обрано (рис. 3.24).

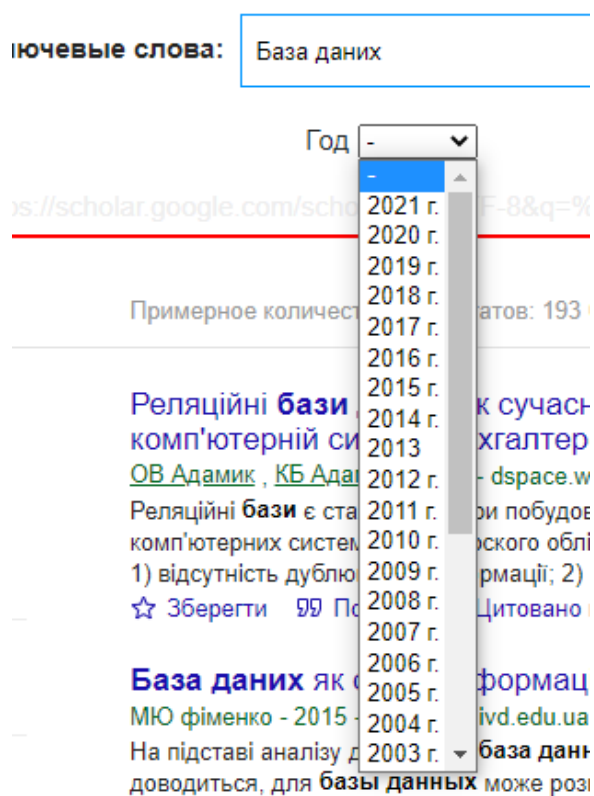
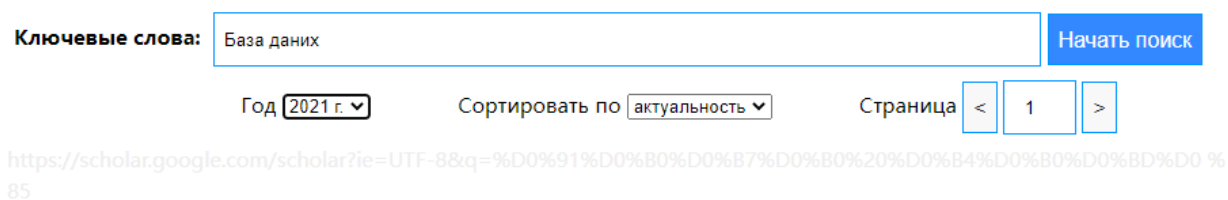


Рисунок 3.22 – Результат роботи розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань



Идет загрузка ... Подождите 5 сек.

© 2021 [Дипломная работа](#)

Рисунок 3.23 – Очікування результату сортування за роком

Примерное количество результатов: 14 900 (0,06 сек.)

Також розроблений програмний продукт дозволяє здійснювати аналіз динаміки розвитку досліджень в окресленій галузі (рис. 3.25). Графічне представлення результатів вибірки у певній науковій сфері досліджень дозволяє об'єктивно оцінювати сучасні наукові набутки за сферою знань.



Рисунок 3.25 – Результат роботи розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

Під час тестування збоїв та недоліків у роботі програми не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

Висновки до розділу 3

У межах третього розділу здійснено проектування й розробку інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. Розроблено діаграму використання, діаграму компонентів та діаграму класів. Описано основні потоки, що циркулюють у системі.

4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ СИСТЕМИ ПОШУКУ ДОСЛІДЖЕНЬ В ГАЛУЗІ КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

4.1 Підготовка до експерименту

На сьогоднішній день, одним з головних напрямків ефективної реалізації будь-якої методики є експериментальне дослідження тобто виявлення якостей досліджуваних об'єктів, перевірка достовірності гіпотез, а також широке та глибоке вивчення досліджуваної наукової тематики. У рамках сучасної науки існує багато різних класифікацій експериментів в залежності від галузі науки, мети дослідження, структури об'єктів та явищ, організаційних заходів, характеру взаємодії об'єкту та засобів дослідження тощо.

Провідним місцем у досконалому проведенні експерименту, займає правильна розробка методики експерименту – визначена послідовність процесів, у результаті якої досягається мета дослідження.

Першочерговим етапом проведення експериментального дослідження є план програми дослідження, який складається за умови проведення дослідження, де визначається:

- гіпотеза;
- мета та задачі;
- вхідні і вихідні параметри, область їх визначення та крок дискредитації;
- порядок проведення власне експерименту;
- зазначаються необхідні засоби проведення дослідження, моделювання, обробки результатів;
- порядок і вимоги щодо оформлення результатів.

Далі, слідує етап визначення об'єму експериментальних досліджень та необхідних програмних та апаратних засобів тощо.

Останнім кроком є безпосередньо експеримент, який проводиться з регламентацією всіх кроків, та обробка і систематизація експериментальних і усіх числових даних, перевірка зведення до єдиної системи одиниць, побудова графіків залежностей, таблиць, діаграм тощо.

Гіпотеза

Експеримент базується на припущенні, що запропонована система пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань адекватно реагує на зміну ідентифікуючих та оціночних параметрів при різних умовах контрольованого середовища.

Мета та задачі експерименту

Метою експерименту є перевірка адекватності запропонованої системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, а саме:

- дослідження запропонованої системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань на основі експертних методів та нечіткої логіки стосовно ефективності її роботи;

- дослідження запропонованої системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань на основі експертних методів та нечіткої логіки стосовно ефективності її роботи, а саме коректності оцінювання критичності ситуації, що склалася під впливом роботи.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- обробка і верифікація отриманих результатів;
- проведення виявлення недоліків та оцінки критичності ситуацій в контрольованому середовищі при зміні ідентифікуючих та оціночних параметрів
- визначення можливості використання системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань для проведення аналізу досліджень.

Вибір вхідних та вихідних параметрів

Для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

- вхідні параметри – значення параметру, KB1, Пар23, KB2,, Пар Н, KB Н, а також порівняльні судження експертів, щодо важливості оціночних е -го та е'-го параметрів між собою відповідно до методу кількісного парного порівняння з визначенням квадратного кореня;
- вихідні параметри – показник рівня критичності ситуації, спричиненої впливом роботи, відображений індикатором критичності.

Послідовність дій в експериментальному дослідженні

Для дослідження системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань виконуються в повній відповідності до етапів методу оцінки критичності ситуації та режиму роботи системи – задається множина оціночних параметрів; визначаються KB важливості кожного параметра; проводиться фазифікація їх поточних значень; обчислюється показник рівня критичності ситуації; отриманий показник порівнюється з оціночним еталоном, на основі чого приймається рішення щодо критичності ситуації; проводиться дефазифікації значень оціночних параметрів та рівня критичності і на основі отриманих даних створюється індикатор критичності.

Засоби проведення експерименту

Для дослідження, створення необхідного програмного забезпечення, імітаційного модулювання, обробки результатів та представлення їх в табличному та графічному вигляді використовувалося середовище Microsoft Excel 2007.

Аналіз результатів

Аналіз результатів імітаційного моделювання буде представлено у підрозділі 4.3 даної роботи. Результати представлені в табличній формі та у вигляді графіків і діаграм.

4.2 Проведення експерименту

Проведемо порівняння розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань з аналогами наведеними у першому розділі даної роботи.

У якості параметрів досліджуються оцінки експертів щодо ефективності, параметри наведені у таблиці 4.1. Далі на основі отриманих даних проводиться математичний аналіз та здійснюється побудова діаграм та графіків для детального розуміння ефективності розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

Таблиця 4.1 – Параметри звернень інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

Параметр	Опис	Можливі значення
$E_{\text{ф}}$	Ефективність роботи	5-надвисока,..., 0-низька
$I_{\text{н}}$	Інноваційність розробки	1- інновація, 0-стандарт
$O_{\text{б}}$	Обсяги даних, що надходять на обробку	10-макс,..., 0-мінім.
$K_{\text{д}}$	Категорії даних, що надходять на обробку	1,2,3...n - категорія
$P_{\text{п}}$	Пріоритет даних, що надходять на обробку	1,2,3...n - пріоритет
$P_{\text{кін}}$	Кінцевий рівень вирішення завдання	1,2,3,4 - рівні
$C_{\text{звер}}$	Стан дієвості	0-відкритий, 1-закритий
K	Коефіцієнт якості розробки	5-надвисокий, ..., 0-низький
$O_{\text{клієн}}$	Оцінка клієнта розробки	0,1,2,3,4,5

У табл. 4.1 показано параметри звернень, що надходять, їх опис та можливі значення розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

4.3 Результати експерименту

Далі складемо таблицю значень отриманих при дослідженні розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань та аналогів, що прийняті до розгляду.

Таблиця 4.2 – Значення показників діяльності розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

Параметр	Значення									
E_{ϕ}	5	5	3	5	3	5	4	5	4	5
ІН	1	0	2	1	0	1	2	1	0	1
O_b	8	10	9	8	9	10	7	8	9	10
K_d	1	2	4	1	3	2	4	1	3	2
P_{Π}	2	5	8	8	2	2	5	2	3	5
$P_{\text{кін}}$	1	2	3	1	3	2	4	1	3	2
$C_{\text{звер}}$	1	0	1	1	0	1	1	3	0	1
K	5	5	5	5	3	5	5	5	5	3
$O_{\text{клієн}}$	5	5	4	4	5	5	5	5	5	5

Таблиця 4.3 – Значення показників діяльності Google Scholar

Параметр	Значення									
Е _ф	2	2	3	2	3	2	3	2	3	2
ІН	1	1	1	1	1	1	1	1	1	1
О _б	3	4	3	4	4	2	7	8	3	2
К _д	1	2	2	1	3	2	2	1	3	2
П _п	2	5	2	2	2	1	5	2	3	2
Р _{кін}	1	2	3	1	3	2	2	1	3	2
С _{звер}	1	1	1	1	1	1	1	1	1	1
К	2	2	4	2	3	2	2	2	2	4
О _{клієн}	2	2	4	4	2	2	2	2	3	3

Таблиця 4.4 – Значення показників діяльності Dimensions

Параметр	Значення									
Е _ф	3	2	3	2	3	2	3	3	3	2
ІН	1	1	1	1	1	1	1	1	1	1
О _б	5	4	3	4	4	2	7	2	3	5
К _д	2	2	2	1	3	2	2	1	3	2
П _п	5	1	2	2	5	1	5	2	3	2
Р _{кін}	3	2	3	1	2	2	2	1	2	2
С _{звер}	1	1	1	1	1	1	1	1	1	1
К	4	2	4	3	3	2	3	2	4	3
О _{клієн}	4	2	4	4	2	2	4	2	3	4

Відповідно до наведених даних здійснюємо аналіз отриманих даних. Розраховуємо середньоквадратичне відхилення та дисперсію, отримані результати наводимо у таблиці 4.5.

Таблиця 4.5 – Результати математичного аналізу

	Комп'ютерна система для інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань		Google Scholar		Dimensions	
	Середнє відхилення	Дисперсія	Середнє відхилення	Дисперсія	Середнє відхилення	Дисперсія
E_{ϕ}	0,84	0,81	0,48	0,24	0,54	0,49
ІН	0,42	0,21	0,48	0,24	0,48	0,24
O_{ϕ}	0,84	0,96	1,4	3,6	1,2	2,4
K_d	0,84	0,96	0,54	0,49	0,9	1,05
P_{Π}	2,1	5,69	1,04	1,64	1,28	2,09
$P_{\text{кін}}$	0,84	0,96	0,6	0,6	0,82	1,01
$C_{\text{звер}}$	0,42	0,21	0,48	0,24	0,5	0,25
К	0,56	0,44	0,7	0,65	0,54	0,49
$O_{\text{клієн}}$	0,32	0,16	0,72	0,64	0,72	0,64

Наступним кроком є кореляційний аналіз. Результати у таблиці 4.6.

Таблиця 4.6 – Кореляційний аналіз отриманих результатів розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

	1									
E_{ϕ}	0,911697	1								
ІН	0,617393	0,868558	1							
O_{ϕ}	0,728198	0,881269	0,898442	1						
K_d	0,832953	0,894452	0,728182	0,597219	1					
Π_{Π}	0,962961	0,885368	0,582819	0,618258	0,911515	1				
$P_{\text{кін}}$	0,748442	0,921121	0,857917	0,766199	0,881287	0,751921	1			
$C_{\text{звер}}$	0	0,911697	0,617393	0,728198	0,832953	0,962961	0,748442	1		
К	0,864767	0,937469	0,785538	0,693577	0,966478	0,916413	0,941168	0,864767	1	
$O_{\text{клієн}}$	0,912829	0,985191	0,867125	0,874917	0,891533	0,884572	0,861267	0,912829	0,912937	1
	E_{ϕ}	ІН	O_{ϕ}	K_d	Π_{Π}	$P_{\text{кін}}$	$C_{\text{звер}}$	К	$O_{\text{клієн}}$	

Таблиця 4.7 – Кореляційний аналіз отриманих результатів діяльності Google Scholar

	1									
Е _ф	0,815432	1								
ІН	0,739299	0,372583	1							
О _б	0,91632	0,638242	0,754777	1						
К _д	0,815627	0,653376	0,714917	0,632127	1					
П _п	0,575396	0,216284	0,811931	0,562544	0,83666	1				
Р _{кін}	0,833118	0,887345	0,295395	0,689483	0,662137	0,235712	1			
С _{звер}	0,775117	0,597659	0,355539	0,736969	0,611845	0,359918	0,848311	1		
К	0,768293	0,759345	0,671169	0,677148	0,864638	0,714213	0,618661	0,415517	1	
О _{клієн}	0,615312	0,324185	0,887125	0,565811	0,55814	0,628619	0,148167	0,168168	0,494416	1
	Е _ф	ІН	О _б	К _д	П _п	Р _{кін}	С _{звер}	К	О _{клієн}	

Таблиця 4.8 – Кореляційний аналіз отриманих результатів діяльності Dimensions

	1									
Е _ф	0,51521	1								
ІН	0,679674	0,681919	1							
О _б	0,852941	0,715124	0,831611	1						
К _д	0,83247	0,38651	0,382633	0,568954	1					
П _п	0,379628	0,875	0,811931	0,596559	0,276179	1				
Р _{кін}	0,939549	0,645159	0,565217	0,859587	0,826324	0,415465	1			
С _{звер}	0,686683	0,448833	0,654812	0,696932	0,678262	0,472456	0,661768	1		
К	0,761523	0,589925	0,785312	0,753211	0,789211	0,6742	0,691867	0,777212	1	
О _{клієн}	0,766136	0,828125	0,727995	0,942292	0,483138	0,625	0,847791	0,51789	0,623635	1
	Е _ф	ІН	О _б	К _д	П _п	Р _{кін}	С _{звер}	К	О _{клієн}	

Таким чином, на основі проведеного математичного аналізу ефективності роботи розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань та використовуючи шкалу Чеддока, можна зробити висновок, що найбільш впливають на ефективність такі чинники:

Для розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань:

- ефективність (E_{ϕ});
- пріоритет ($\Pi_{\text{п}}$);
- категорія даних ($K_{\text{д}}$);
- коефіцієнт якості (K);
- оцінка клієнта ($O_{\text{клієн}}$).

Для Google Scholar:

- ефективність (E_{ϕ});
- обсяги даних ($O_{\text{б}}$);
- категорія даних ($K_{\text{д}}$);
- кінцевий рівень вирішення ($P_{\text{кін}}$).

Для Dimensions:

- обсяги даних ($O_{\text{б}}$);
- категорія даних ($K_{\text{д}}$);
- кінцевий рівень вирішення ($P_{\text{кін}}$).

Далі до розгляду приймаємо: застосунок для інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань та найбільшого конкурента Google Scholar, так як інші програми виявилися не конкурентоспроможними за результатами попереднього аналізу.

На виході цього етапу, згідно до формули:

$$KPI = \{\cup_{w=1}^v KPI_w\} = \{KPI_1, KPI_2, \dots, KPI_v\},$$

де $KPI_w \subseteq KPI, (w = \overline{1, v})$

v – кількість ключових показників ефективності.

Для застосунку інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань при $w = 5$ отримуємо множину ключових показників ефективності KPI :

$$KPI = \{E_\phi, \Pi_\pi, K_d, K, O_{\text{клієн}}\} = \left\{ \begin{array}{l} \text{ефективність, пріоритет, категорія даних,} \\ \text{коефіцієнт якості, оцінка клієнта} \end{array} \right\}$$

Для Google Scholar при $w = 4$ отримуємо множину ключових показників ефективності KPI :

$$KPI = \{E_\phi, O_\phi, K_d, P_{\text{кін}}\} = \left\{ \begin{array}{l} \text{категорія даних, обсяги даних,} \\ \text{ефективність,} \\ \text{кінцевий рівень звернень} \end{array} \right\}$$

Далі проведемо графічне порівняння двох програм за основними показниками для виявлення найбільш ефективної та досконалої програми.

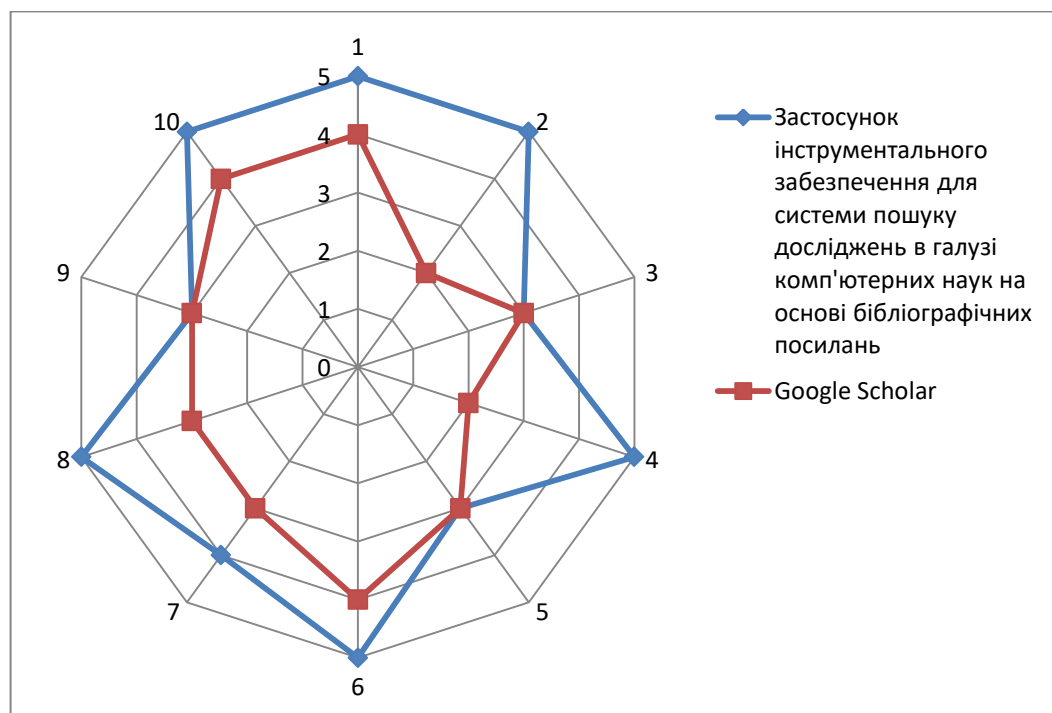


Рисунок 4.1 – Графік ефективності програм

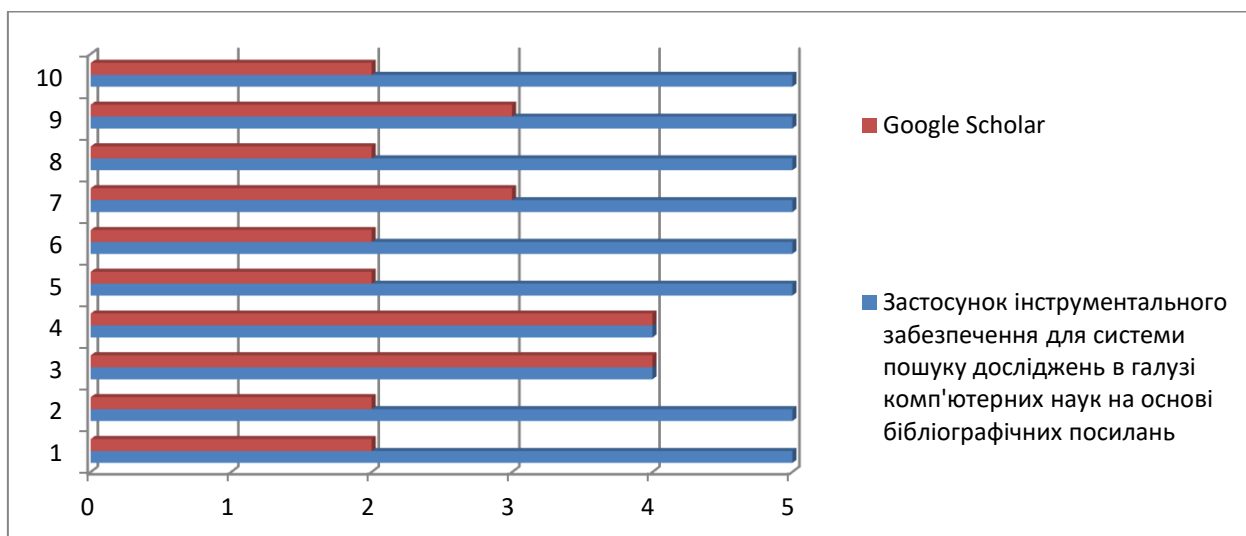


Рисунок 4.2 – Графік оцінка клієнта програм

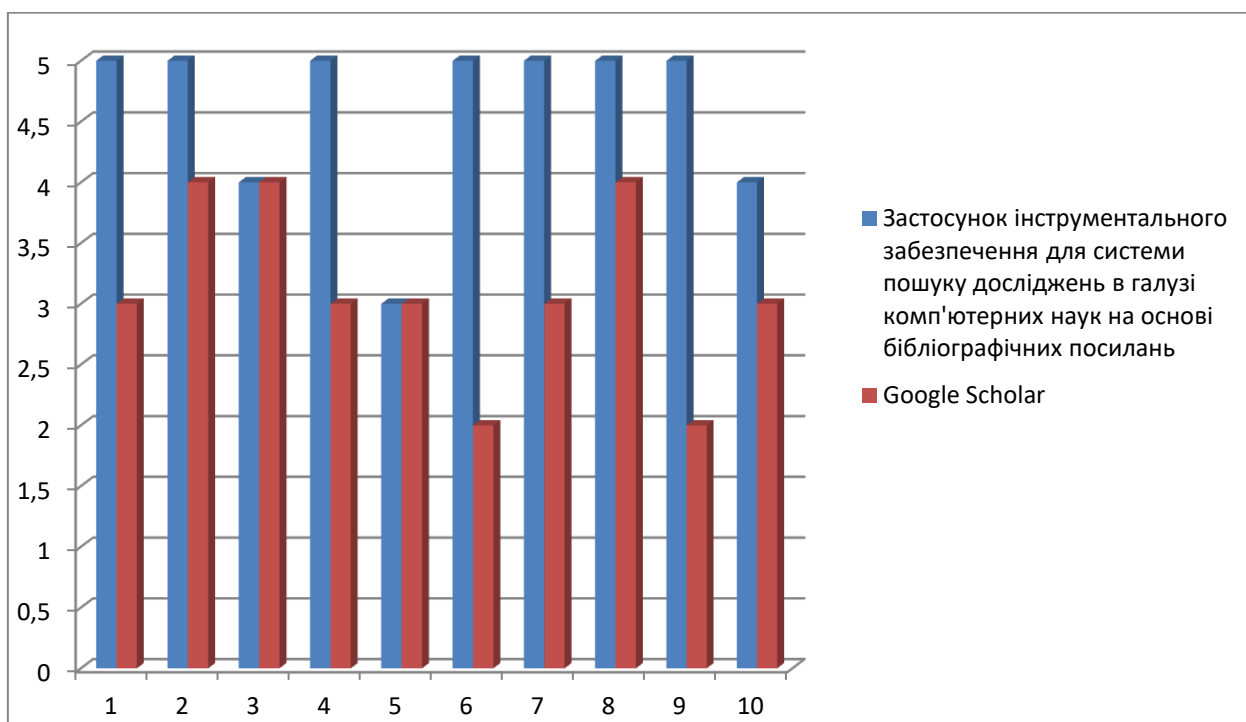


Рисунок 4.3 – Графік коефіцієнт якості програм

Таким чином, згідно до проведеного дослідження варто відзначити, що застосунок інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань за всіма показниками перевершує своїх конкурентів/аналогів, що говорить про високу якість та ефективність програми та можливість впровадження в реальну роботу за потребою.

Висновки до розділу 4

У рамках четвертого розділу здійснено дослідження розробленого інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань та проведено порівняння з аналогами наведеними у першому розділі даної роботи.

У результаті дослідження ефективності варто відзначити, що застосунок інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань за всіма показниками перевершує своїх конкурентів/аналогів, що говорить про високу якість та ефективність програми та можливість впровадження в реальну роботу за потребою.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Вимоги безпеки при виконанні робіт на робочому місці

У розділі розглядається тема охорони праці та безпеки працівників у надзвичайних ситуаціях. Мета даної частини роботи – є аналіз стану умов праці середовища, де розроблювалась програма дипломної роботи. Для розв’язання питання буде використано норми України.

На робочому місці працівника можуть статися ситуації, які призведуть на нього негативний вплив пов’язаний зі станом його психічного та фізичного здоров’я. Робота з комп’ютером не є цілком безпечною, через високий рівень напруженості праці. Важливою складовою роботи є правильно організоване робоче місце, що попередить погіршення стану людини через монотонність роботи та обмеженість його руху.

Для забезпечення безпеки праці інженера-програміста при роботі з технікою існують нормативно-правові акти затверджені та чинні в Україні. При виконанні норм на робочому місці життєдіяльність працівників покращується, стає більш безпечною. Розглядалися наступні акти:

- НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» затверджені наказом Міністерства соціальної політики України від 14 лютого 2018 №207 [46];
- ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин затверджені постановою Головного державного санітарного лікаря України від 10.12.1998 № 7 [47];
- Примірня інструкція з охорони праці під час експлуатації електронно-обчислювальних машин, затверджена наказом Міністерства доходів і зборів України від 05.09.2013 № 443 [48].

5.2 Шкідливі фактори на робочому місці

Шкідливий виробничий фактор - фактор середовища і трудового процесу, вплив якого на працюючого за певних умов (інтенсивність, тривалість та ін.) може викликати професійне захворювання, тимчасове або стійке зниження працездатності, підвищити частоту соматичних і інфекційних захворювань, призвести до порушення здоров'я нащадків [49].

Основні шкідливі фактори на робочому місці з якими може стикнутися інженер-програміст перелічені нижче:

- підвищена температура поверхонь комп'ютеру;
- висока яскравість екранів;
- брак світла або його відсутність;
- обмеження рухомості;
- недотримання норм ергономічних показників робочого місця;
- підвищений рівень електромагнітного випромінювання;
- підвищений рівень ризику виникнення пожеж;
- підвищений рівень ризику отримання електротравми;
- нервово-емоційне напруження;
- монотонність праці.

Одним з основних факторів, що впливає на працездатність людей, що використовують комп'ютер у своїй роботі – є безпечна умова праці. Умови праці з комп'ютером визначаються:

- організацією робочого місця;
- умовами середовища перебування працівника (світло, мікроклімат, шум та вібрація, тощо);
- взаємодією людини з комп'ютером.

Далі у підрозділах будуть детально розглянуті шкідливі фактори пов'язані з освітленням, мікрокліматом, організацією робочого місця, шумом та вібрацією.

5.1.1 Характеристики робочого місця

Робоче місце повинно бути у відповідності нормам охорони праці. Приміщення, де була розроблена програма має наступні характеристики:

- одна працююча людина;
- довжина приміщення 5 м;
- ширина приміщення 3 м;
- висота приміщення 2.5 м.

За вхідними даними, можна розрахувати, що загальна площа на одне робоче місце 15 м^2 , а об'єм приміщення – 37.5 м^3 .

Відповідно до документу [47] обсяг приміщення повинен бути не менше 20 м^3 на одну людину, а площа приміщення не менш ніж 6 м^2 . Тож приміщення відповідає стандартам та є придатним для використання.

5.1.2 Освітлення

Штучне освітлення не впливає негативно на стан працівника за умови його правильного оформлення. Це покращує умови праці, знижує стомлюваність, підвищує продуктивність праці, позитивно впливає на робоче середовище, позитивно впливає на працівника, підвищує безпеку та зменшує травматизм.

Якщо освітлення на робочому місці оформлено неправильно або неадекватно, людина може відчувати втому очей, втрату уваги і передчасну втому під час роботи. Занадто яскраве світло призводить до сліпоти, роздратування і сліз в очах. Неправильний напрямок світла на робочому місці може призвести до різких тіней, відблисків і дезорієнтації працівника. Всі ці причини можуть призвести до нещасних випадків або професійних захворювань, тому важливо правильно розрахувати освітленість.

Освітлення в приміщеннях регламентується ДБН В.2.5-28-2018 [50]. Під час роботи використовувалося природне одностороннє та штучне освітлення. У зв'язку з тим, що категорія візуальної роботи на дисплеї ПК відноситься до категорії III, освітленість робочого місця при загальному

освітленні повинна бути від 200 до 400 лк. Фактична освітленість на робочому місці 191,8 лк.

Відповідно до ДБН В.2.5-28-2018 [50] в приміщенні використовується природне та штучне освітлення. Отже, наявних джерел світла достатньо для роботи з дисплеєм.

5.1.3 Мікроклімат

Нормативи параметрів мікроклімату промислових підприємств наведено в ДСН 3.3.6.042-99 [51]. Мікроклімат виробничого середовища впливає на стан організму працівника, істотно змінюючи його працездатність протягом робочого дня. Показники температури, відносної вологості, швидкості руху повітря, теплового випромінювання від нагрітих поверхонь характеризують клімат внутрішнього середовища виробничого приміщення. Нижче наведені основні показники мікроклімату робочого простору і оптимальні норми, в межах яких вони повинні розташовуватися:

- температура повітря в робочому приміщенні в холодну пору року становить 20 °С, а в теплу 24 °С (при оптимальних показниках в холодну пору року - 16 - 24 °С; в теплу - 18 - 25 °С);

- відносна вологість повітря знаходиться в межах 45 - 55% (при оптимальних значеннях 40 - 60%);

- швидкість повітря - в холодну пору року 0,2 м/с (при швидкості 0,1-0,3 м/с) і в теплу пору року 0,4 (при швидкості 0,2-0,4 м/с).

Показники мікроклімату в приміщенні, де була розроблена програма для дослідження, порівнювалися за холодною пори року. Показання в кімнаті такі:

- температура повітря в приміщенні: 21 °С;
- відносна вологість: 40%;
- швидкість повітря: 0 м/с.

У кімнаті два джерела тепла: ПК та система опалення з радіаторами. Всі показники відповідають нормам в холодну пору року.

5.1.4 Шум та вібрація

Надмірний шум і вібрація впливають та завдають шкоди робочому процесу людини. Працівники, які працюють у шумних умовах, зазвичай відчують головні болі, дратівливість, втому, біль у вухах тощо. Погіршується не тільки здоров'я людини. Шум і вібрація можуть ускладнювати спілкування працівників в одному приміщенні.

Рівні шуму на виробництві регулюються наказом МОЗ «Про затвердження Державних санітарних норм допустимих рівнів шуму в приміщеннях житлових та громадських будинків і на території житлової забудови» від 22 лютого 2019 р. № 463 [52]. Рівень звуку в приміщенні 40 дБ (при допустимій нормі 50 дБ), критерій шуму 40, що відповідає допустимій нормі.

5.3 Дії працівників в надзвичайних ситуаціях

Існує багато типів надзвичайних ситуацій, з якими інженер-програміст може зіткнутися на робочому місці, включаючи ураження електричним струмом.

Організаційні заходи для досягнення безпеки роботи в електроустановках наводяться у документі [55]:

- затвердження переліку робіт, що виконуються за замовленнями, наказами та в порядку поточної експлуатації;
- призначення відповідальних за безпеку праці;
- оформлення робіт за нарядом, замовленням чи погодженням переліку робіт, що виконуються в порядку поточної експлуатації;
- підготовка робочих місць;
- допуск до роботи;
- нагляд під час виконання робіт;
- переведення на інше робоче місце;

— оформлення перерв у роботі та її припинення.

На підставі наказу МОЗ України «Порядок надання домедичної допомоги постраждалим при ураженні електричним струмом та блискавкою» [53] розроблено інструкцію щодо дотримання у разі нещасного випадку:

- переконайтеся, що немає небезпеки. негайно вимикніть електроінструмент, портативні прилади та пристрої. Повідомте керівника;
- якщо потерпілий перебуває під дією електричного струму, по можливості припиніть дію: вимкніть джерело живлення, викиньте електричний кабель сухою дерев'яною паличкою або іншим не струмопровідним засобом;
- у разі тяжкого випадку викликати швидку допомогу;
- якщо потерпілий втратив свідомість, його необхідно рівно розмістити, розстебнути одяг, дати йому подихати свіжим повітрям.
- якщо постраждалий не дихає, до приїзду швидкої необхідно зробити штучне дихання та масаж серця;
- накладіть чисті стерильні пов'язки при опіках;
- у разі погіршення стану постраждалого до приїзду швидкої необхідно повторно викликати керівника швидкої допомоги.

Ні в якому разі не можна відмовлятися від надання першої допомоги, навіть якщо потерпілий здається мертвим через задишку, серцебиття та інші життєво важливі показники.

Для безпечної роботи у пожежонебезпечному приміщенні необхідно дотримуватися таких правил документа «Про затвердження Правил пожежної безпеки» [54].

Заходи, які необхідно вжити у разі виникнення пожежі на робочому місці, включають:

- звернення до пожежної охорони за номером 101, увімкніть пожежну сигналізацію. За потреби викликати інші оперативні служби;
- повідомити керівництво про пожежу;
- вживати заходів щодо гасіння пожежі вогнегасником;
- евакуюйтесь з приміщення в безпечну зону. Якщо утворюється дим, краще повзти і, по можливості, прикласти до дихальних шляхів вологу тканину.
- відчиняючи двері, переконайтеся, що ручка дверей не тепла, а за дверима немає вогню. Завжди зачиняйте двері, щоб зменшити поширення вогню;
- не намагайтеся брати з собою майно, краще евакуюватися раніше.

Висновки до розділу 5

У цьому розділі аналізуються документи з безпеки на робочому місці, відповідності характеристик приміщення, в якому розроблена програма, стандартам, затвердженим державними органами.

У результаті можна констатувати, що робоче місце відповідає вимогам безпеки та гігієни праці. Під час розрахунків було підтверджено, що освітлення, шум і вібрація, мікроклімат знаходяться в межах норми.

З метою забезпечення безпеки в аварійних ситуаціях, коли сталося ураження електричним струмом, підготовлено інструкцію для екстрених дій для догляду постраждалих. Також розроблено план заходів з евакуації з пожеж на робочих місцях.

ВИСНОВКИ

У ході виконання даної магістерської роботи здійснено розробку пошукового сервісу наукових публікацій з ціллю аналізу динаміки розвитку досліджень в галузі комп'ютерних наук на основі знайдених бібліографічних посилань.

Для цього у роботі:

- розкрито експериментальний метод досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, визначено його дієвість та ефективність;
- здійснено проектування й розробку інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань;
- проведено дослідження ефективності системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

На основі вищевикладеного варто зробити наступний висновок:

У загальному випадку результати пошуку насамперед залежать від поставленого завдання та коректності запиту. Але ці результати найчастіше, з одного боку, надмірні та з іншого боку неповні.

Проте, і автори та видавці, як правило, зацікавлені в тому, щоб інформація про публікації індексувалася пошуковими системами, але тут є нюанси: не завжди дозволяється індексація вмісту pdf-файлів, і в деяких випадках дозволена індексація сайтів лише певними пошуковими системами.

Окрім іншого, результати запиту залежать від порядку слів та від IP-адреси, з якої здійснюється пошук.

Пошук інформації в мережі – це досить специфічна та кропітка робота, яка потребує певних знань та навичок. Для проведення пошукових робіт знадобляться: знання основних інформаційних ресурсів і вміння добре в них орієнтуватися, практичні навички роботи.

Розглянуто алгоритм функціонування системи веб-пошуку, яка розглядається у межах даної дипломної роботи. Наведено блок-схеми алгоритму функціонування системи та сформовано перелік завдань які вирішує дана система.

Проведено експериментальне дослідження роботи запропонованого методу. Отримані результати показали, що метод успішно справляється з поставленим завданням. Проте в ході дослідження були виявлені проблеми, вирішення яких є подальшим розвитком методу і, швидше за все, дозволить досягти кращих результатів роботи.

Розроблений програмний продукт дозволяє здійснювати аналіз динаміки розвитку досліджень в окресленій галузі. Під час тестування збоїв та недоліків у роботі програми не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

Застосунок інструментального забезпечення для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань за всіма показниками перевершує своїх конкурентів/аналогів, що говорить про високу якість та ефективність програми та можливість впровадження в реальну роботу за потребою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Артюхова Ю. І., Мочалова Н. С. Аналіз методів розширення запитів інформаційного пошуку як засобів збільшення пошукових можливостей / XV Всеукраїнська науково-методична конференція студентів і молодих науковців «Прикладна лінгвістика 2019: проблеми та рішення», 2019. – <http://eir.nuos.edu.ua/xmlui/bitstream/handle/123456789/2657/Mochalova.PDF?sequence=1>
2. Поморова О. В. Модуль обробки текстових документів автоматизованої інформаційної системи зберігання та опрацювання наукових робіт [Текст] / О. В. Поморова, Л. О. Ковтун, В. Б. Белза // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2018. – № 2. – С. 87-92.
3. Матвійчук Т. Методи інформаційного пошуку. VIII Всеукраїнська студентська науково-технічна конференція «Природничі та гуманітарні науки. Актуальні питання». С. 116–117.
4. Сухий О. Л., Міленін В. М., Тарадайнік В. М. Алгоритми пошуку в інформаційних системах : методичні рекомендації. К., 2015.
5. Шокин Ю. И., Федотов А. М., Барахнин В. Б. Проблемы поиска информации. Новосибирск: Наука, 2010. 220 с.
6. Максимов Н. В., Голицына О. Л., Тихомиров Г. В., Храмцов П. Б. Информационные ресурсы и поисковые системы. Учебное пособие. Москва: МИФИ, 2008.
7. Chowdhury G. G. Information Retrieval, 3rd edition. London: Facet Publishing, 2010. 488 p.
8. Гендина Н. И. Информационно-поисковые тезаурусы: структура, назначение и порядок разработки. – 14 с. URL: <https://nsu.ru/xmlui/bitstream/handle/nsu/8962/IPT.pdf>
9. Полянко А. Особенности поисковой системы Google [Електронний ресурс] / А. Полянко // Siteseo.ru – поисковая оптимизация и интернет-маркетинг – Режим доступа : <http://siteseo.ru/info/seostati/GoogleMSNYa/40/>.

10. История поисковой системы Rambler или особенности Rambler [Электронный ресурс] // Seoalabis.ru – Search engine optimization. – Режим доступа : http://seo-alabis.ru/seo/rambler_istoriya.php.
11. Scirus – for scientific information only [Электронный ресурс] // Scirus – for scientific information only. – Режим доступа : <http://www.scirus.com>.
12. Кутовенко А. Научный интернет-поиск / А. Кутовенко // Мир ПК – 2010. – № 5. – С. 64–67. – ISSN 0235-3520.
13. Максимов Н. В. Информационные ресурсы и поисковые системы / Н. В. Максимов, О. Л. Голицына, Г. В. Тихомиров, П. Б. Храмцов, М : МИФИ, 2008. – 400 с.
14. Электронный ресурс: Sergey Brin and Larry Page. The Anatomy of a Search Engine <http://wwwdb.stanford.edu/pub/papers/google.pdf>.
15. Сегалович И. В. Как работают поисковые системы // Мир Internet. – 2002. – № 10.
16. Searching the Web. / A. Arasu, [and others] // ACM Trans. on Internet Technology. – 2001. – № 1 (1).
17. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. An Introduction to Information Retrieval, Online edition (c) 2009 Cambridge UP, Draft of April 1, 2009, Website: <http://www.informationretrieval.org>
18. Черний Ю.Ю. Школа наукової інформації. Інформаційні потреби. Основи інформаційного пошуку, <http://www.bogoslov.ru/text/321597.html>
19. Захаров В.П. Информационно-поисковые системы. Учебно-методическое пособие, Санкт-Петербург, 2005.
20. Медведь В.Н. Методы поиска информации, <http://northedu.ru/content/view/115/159/>
21. Van Rijsbergen C.J. Information Retrieval. London: Butterworths, 1979. Available at <http://www.dcs.gla.ac.uk/Keith/Preface.html>
22. Шарапов Р.В., Шарапова Е.В., Саратовцева О.А. Модели информационного поиска.

23. Некрестьянов И.С. Тематико-ориентированные методы информационного поиска: Дис. ... канд. техн. наук. – СанктПетербургский государственный университет. – СПб, 2000. – 88 с.
24. Дубинский А.Г. Некоторые вопросы применения векторной модели представления документов в информационном поиске // Управляющие системы и машины. – 2001. – № 4. – С. 77–83.
25. Когаловский М.Р. Перспективные технологии информационных систем. – М.: ДМК Пресс; М.: Компания АйТи, 2003. – 288 с.
26. Holscher C. and Strube G. Web search behaviour of Internet experts and Newbies. Proceedings of WWW9. 2000. Available at <http://www9.org/w9cdrom/81/81.html>.
27. Navarro-Prieto R., Scaife M. & Rogers Y. Cognitive Strategies in Web Searching. Proceedings of the 5th Conference on Human Factors & the Web, 1999. Available at <http://zing.ncsl.nist.gov/hfweb/proceedings/na-varro-prieto/index.html>.
28. Muramatu J. and Pratt W. Transparent queries: Investigating Users' Mental Models of Search Engines, Proceedings of SIGIR 2001.
29. Choo C. W., Detlor B., and Turnbull D. Information Seeking on the Web – An integrated model of browsing and searching. Proceedings of the Annual Meeting of the American Society for Information Science (ASIS), 1999. Available at <http://choo.fis.utoronto.ca/fis/respub/aisis99/>
30. Broder A. A taxonomy of web search, IBM Research, ACM SIGIR Forum archive. – 2002. – Vol. 36 , Issue 2. – P. 3–10.
31. Лексична база англійської мови WordNet, <http://wordnet.princeton.edu/perl/webwn>
32. Онлайн словник <http://dictionary.cambridge.org/>
33. Онлайн словник <http://www.merriam-webster.com/>
34. Rodnessey J. New Search Engines: The Next Generation of Google Competition, 2009, <http://webupon.com/search-engines/newsearch-engines-the-next-generation-ofgoogle-competition/>

35. Nobles R. The Future Of Search Engine Optimizing, <http://www.searchengineworkshops.com/articles/se-optimization-future.html>
36. Андон Ф.И., Гришанова И.Ю., Резниченко В.А. Semantic Web как новая модель информационного пространства интернет // Проблемы програмування. – 2008. – № 2–3. – С. 417–430.
37. Ezzy E., Search 2.0 vs Traditional Search, 2006, http://www.readwriteweb.com/archives/search_20_vs_tr.php
38. McLoughlin S. Searching on the web; the new breed of search engines, 2009, <http://relativemusings.blogspot.com/2009/05/searching-on-web-new-breed-of-smarter.html>
39. Wolfram S. Wolfram Alpha – computational knowledge engine, 2009 <http://basetechnology.blogspot.com/2009/03/wolframalpha-computational-knowledge.html>
40. Сидоров В. Wolfram Alpha – Computational Knowledge Engine, или Как сложить яблоко с апельсином?, блог, 2009, <http://netler.ru/pc/wolfram.htm>
41. Official Google Blog: Square your search results with Google Squared, <http://googleblog.blogspot.com/2009/06/square-your-search-results-with-google.html>
42. Сидоров В. Google Squared: как успех Wolfram Alpha взбудоражил Google и что из этого вышло?..., блог, 2009, <http://netler.ru/pc/google-squared.htm>
43. Soubbotin D. Summarization, the Answer to Web Search: Interview with Dmitri Soubbotin of SenseBot, Search Engine Journal, 2007, <http://www.searchenginejournal.com/summarization-the-answer-to-web-search-interview-with-dmitri-soubbotin-of-sensebot/6094/>
44. Левшин Д. Web, часть третья // Открытые системы. – 2008. – № 2. <http://cio.ru/text/print/302/8165094.html>
45. Рогушина Ю.В., Гришанова И.Ю. Разработка принципов представления электронных изданий, обеспечивающих корректную индексацию поисковыми системами Интернет // Проблемы програмування. – 2004. – № 4. – С. 39–47.

46. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» затверджені наказом Міністерства соціальної політики України від 14 лютого 2018 №207.
47. ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин затверджені постановою Головного державного санітарного лікаря України від 10.12.1998 № 7.
48. Примірний інструкція з охорони праці під час експлуатації електронно-обчислювальних машин, затверджена наказом Міністерства доходів і зборів України від 05.09.2013 № 443.
49. Про затвердження Гігієнічної класифікації праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу № 528 від 27.12.2001
50. Державні будівельні норми України «Природне і штучне освітлення» ДБН В.2.5-28:2018, затверджені наказом Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України 03.10.2018 № 264, введено в дію з 01.03.2019
51. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042- 99, затверджені Постановою головного санітарного лікаря України № 42 від 1 грудня 1999 року
52. Наказ Міністерства охорони здоров'я «Про затвердження Державних санітарних норм допустимих рівнів шуму в приміщеннях житлових та громадських будинків і на території житлової забудови» від 22 лютого 2019 року №463
53. Наказ про «Порядок надання домедичної допомоги постраждалим при ураженні електричним струмом та блискавкою» зареєстрований в Міністерстві юстиції України 7 липня 2014 р. за № 775/25552.
54. НАПБ А.01.001-2014 Правила пожежної безпеки в Україні, затверджений наказом Міністерства внутрішніх справ України від 30.12.2014 № 1417 і зареєстрований у Міністерстві юстиції України 05.03.2015 за № 252/26697

55. НПАОП 40.1-1.21-98 «Про затвердження Правил безпечної експлуатації електроустановок споживачів» зареєстровано в міністерстві юстиції України 10 лютого 1998 р. за № 93/2533

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Перший проректор Дніпровського
національного університету
залізничного транспорту
імені академіка В. Лазаряна
Борис БОНДАР

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДОСЛІДЖЕНЬ В ГАЛУЗІ
КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

Технічне завдання
ЛИСТ ЗАТВЕРДЖЕННЯ
1116130.01212-01-ЛІЗ

Завідувач кафедри КІТ

доц. Вадим ГОРЯЧКІН



Керівник розробки

доц. Вадим АНДРЮЩЕНКО



Виконавець

студент групи ПЗ2021
Євген САЄНКО



Нормоконтролер

доц. Олена КУРОП'ЯТНИК



2021

ЗАТВЕРДЖЕНО
1116130.01212-01

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДОСЛІДЖЕНЬ В ГАЛУЗІ
КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ**

1116130.01212-01

Аркушів 22

2021

АНОТАЦІЯ

Документ 1116130.01212-01 «Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. Технічне завдання» входить до складу програмної документації дипломного проекту.

У документі представлене призначення та область застосування програмного продукту, основні вимоги, стадії та терміни виконання проекту, техніко-економічні показники, що пред'являються до програмного продукту.

ЗМІСТ

Вступ	5
1 Підстава до розробки	6
2 Призначення розробки	7
2.1 Функціональне призначення	7
2.2 Експлуатаційне призначення	7
3 Вимоги до програми	8
3.1 Вимоги до функціональних характеристик	8
3.2 Вимоги до надійності	9
3.3 Умови експлуатації	9
3.4 Вимоги до складу та параметрів технічних засобів	9
3.5 Вимоги до інформаційної та програмної сумісності	10
3.6 Вимоги до маркування та упаковки	9
3.7 Вимоги до транспортування та зберігання	10
4 Вимоги до програмної документації	11
5 Визначення витрат на проектування програми	12
6 Стадії та етапи розробки	20
7 Порядок контролю та приймання	21
Бібліографічний список	22

ВСТУП

«SchoolMaster» являє собою програмне забезпечення для аналізу динаміки розвитку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. Система доступна як додаток і може функціонувати в операційній системі Windows.

Оскільки основна частина обчислювальних ресурсів зосереджена на етапі збору бібліографічних посилань за ключовим словом, то необхідно розробити швидкодіючу структуру для аналізу динаміки розвитку досліджень та побудові графіку результатів аналізу. Графік аналізу результатів **розвитку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань** повинен бути побудований у двовимірному просторі на головній сторінці додатку. Побудова графіку відбувається поступово за рахунок формування кількості публікацій за певним роком.

Причиною замовлення продукту стала відсутність прямих аналогів на українському ринку, а також неможливість адаптації існуючих аналогів через відсутність відкритого доступу до коду.

Сферою застосування розробленого програмного продукту можуть бути всі наукові галузі, де потрібна попередня обробка досліджень за сферою наукового подання, дослідницькі відділи з аналізу наукових робіт, тощо.

1 ПІДСТАВА ДО РОЗРОБКИ

Підставою для розробки є наказ ректора Дніпровського національного університету залізничного транспорту імені акад. В. Лазаряна Пшінька О. М. №690 ст від 18.11.2020 р. «Про призначення наукових керівників та затвердження тем дипломних проектів» факультету «Комп'ютерні технології та системи» за спеціальністю 121 «Інженерія програмного забезпечення» по кафедрі «Комп'ютерні інформаційні технології».

Тема проекту «Аналіз динаміки розвитку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань», керівник дипломного проекту доц. Андрющенко В.О.

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

2.1 Функціональне призначення

Програмний продукт являє собою пошукову систему, яка формує перелік посилань на дослідження в галузі комп'ютерних наук на основі бібліографічних посилань та здійснює аналіз отриманого списку за роком та за актуальністю, використовуючи отримані дані додаток будує графік на головній сторінці.

2.2 Експлуатаційне призначення

Експлуатаційне призначення програмного засобу:

- застосування в системах обробки бібліографічних посилань для підвищення ефективності їх пошуку у мережі Інтернет;
- застосовування в наукових та видавничих сферах з метою аналізу динаміку розвитку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

3 ВИМОГИ ДО ПРОГРАМИ

3.1 Вимоги до функціональних характеристик

Програмний додаток повинен володіти наступним функціоналом:

- підбір тематичних сайтів,
- пошук інформації на різних ресурсах із різними запитамі,
- завантаження інформації з сайту,
- створення єдиної системи надання інформації користувачеві,
- побудова графіку аналізу досліджень.

Вхідні дані:

- ключові слова за тематикою пошуку;
- обмеження за періодом публікації ресурсів.

Вихідні дані:

- перелік досліджень в галузі комп'ютерних наук на основі бібліографічних посилань;
- графік аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

3.2 Вимоги до надійності

Список вимог до надійності програмного засобу в цьому розділі включає лише пункти, що стосуються програмного забезпечення, збої засобів обчислювальної техніки до нього не входять.

Вимоги до надійності програмного продукту наступні:

- наявність архівної копії тексту програми та файлів графіків на зовнішньому носії інформації;
- не більше ніж одна помилка на тисячу запитів;
- контроль коректності вхідної інформації, яка вводиться користувачем під час роботи з програмним продуктом.

3.3 Умови експлуатації

Даний програмний продукт може використовуватись в умовах, які відповідають вимогам [13].

Для коректного функціонування програмного продукту необхідно дотримання наступних вимог:

- програмне забезпечення повинно використовуватись в приміщеннях, призначених для роботи ЕОМ у наступних кліматичних умовах: температура – 21-25 °С, відносна вологість повітря 40-60%;
- кваліфікація персоналу повинна бути на рівні користувача ЕОМ, що має досвід роботи з операційною системою Windows та пройшов підготовку за керівництвом користувача.

3.4 Вимоги до складу та параметрів технічних засобів

Мінімальна конфігурація комп'ютеру для забезпечення роботи програмного продукту:

- комп'ютер з тактовою частотою процесора 2 ГГц і розрядністю 64-біти;
- ОЗП об'ємом не менше 6 ГБ;
- вільний дисковий простір 1 ГБ;
- наявність CD/DVD приводу або USB роз'єму для встановлення ПЗ;
- монітор з роздільною здатністю екрану 1024x768 та більше;
- стандартні клавіатура та маніпулятор «миша».

3.5 Вимоги до інформаційної та програмної сумісності

Для функціонування програмного продукту необхідна Windows 7 або більш пізня версія.

3.6 Вимоги до маркування та упаковки

Програма може зберігатися на постійних запам'ятовуючих носіях (CD/DVD-диски, флеш-пристрої). Упаковка продукту повинна забезпечувати захист від механічних пошкоджень та мати маркування, зображене на рис. 3.1.

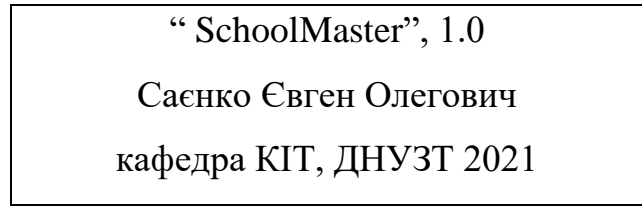


Рис. 3.1. Штамп продукту

3.7 Вимоги до транспортування та зберігання

Умови транспортування та зберігання повинні забезпечувати захист носія від фізичних пошкоджень і відповідати вимогам [2].

Термін зберігання програмного продукту необмежений та залежить лише від умов зберігання та встановленого замовником періоду експлуатації.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація повинна включати наступні документи: технічне завдання та робочий проект у складі:

- специфікація;
- текст програми;
- опис програми;
- керівництво користувача. Керівництво дослідника;

Вся документація до програми повинна задовольняти вимогам державного стандарту до оформлення програмних документів (ГОСТ 19.101-77).

5 ВИЗНАЧЕННЯ ВИТРАТ НА ПРОЕКТУВАННЯ ПРОГРАМИ

Потрібно розробити комплекс програмних засобів для аналізу динаміки розвитку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань та побудови графіку аналізу «SchoolMaster».

Метою даного розділу є розробка техніко-економічного обґрунтування для розроблення нового програмного забезпечення заради оцінки запланованих витрат, а також дати висновок щодо доцільності його розробки та розгортання.

Першим кроком є розрахунок обсягу затрачених трудових ресурсів розробників завдяки оцінці розміру програмного коду. Основні методики визначення запланованих трудовитрат орієнтуються на кількісну або якісну оцінку.

Constructive Cost Model (COCOMO) [1] орієнтується на вимірюваний розмір проекту S в рядках коду LOC (KLOC), а обсяг трудових ресурсів у людино-місяцях, розрахунок наведено у формулі 5.1.

$$E = a \cdot S^b \cdot EAF, \quad (5.1)$$

де E – необхідні людські ресурси на розробку проекту (в людино-місяцях);

S^b – обсяг коду (в KLOC);

EAF – фактор уточнення витрат (effort adjustment factor).

Для простих систем $a = 2,4$; $b = 1,05$.

Для визначення загальної кількості рядків коду можна скористатися засобами Notepad++, що є безкоштовною платформою для розробки програмного забезпечення, в якому проходила розробка всього комплексу програм, та підрахувати довжину всіх програмних модулів. Таким чином визначено, що загальна довжина коду програмного забезпечення становить 1656 рядків. Трудовитрати у людино-місяцях розраховано у формулі 5.2.

$$E = 2,4 \cdot 1,656^{1,05} \cdot 1 = 4,07 \quad (5.2)$$

Далі наведені розрахунки для обчислення вартості розробки програмного комплексу «SchoolMaster». За передбачувані напрямки витрат прийняті:

- грошовий оклад та премії працівників;
- відрахування на соціальні потреби;
- накладні витрати;
- витрати на персональний комп'ютер та ліцензійні програмні засоби, що застосовуються при розробці.

Оцінка основної заробітної плати (ОЗП) інженера-програміста, що займається розробкою програмного продукту, базується на кількості розробників, заробітної плати в перерахунок на години та тривалості розробки (у годинах). Розрахунок заробітної плати відбувається за даними середньої заробітної плати програміста в м. Дніпро [2] і представлений у табл. 5.1.

Таблиця 5.1 – Фонд місячної заробітної плати

№п /п	Посада виконавця	Оклад, грн/міс	Кількість		Сума зарплати грн.
			чол.	місяців	
1	інженер-програміст	17500	1	4,85	84875

Розробка представленого у цій роботі програмного продукту «SchoolMaster» буде проводитися одним програмістом в період з 01.07.2021 до 25.11.2021, що становить 21 робочий тиждень або 105 днів. Робочий час на тиждень приймається за 40 годин. Погодинна оплата кваліфікованого інженера-програміста становить 104,17 грн/год. Таким чином, витрачений робочий час за форм. 5.3 розраховано у форм. 5.4:

$$t_{\text{розробки}} = N_{\text{чол}} \cdot N_{\text{тиж}} \cdot N_{\text{год}}, \quad (5.3)$$

де $N_{\text{чол}}$ – кількість виконавців, чол;

$N_{\text{тиж}}$ – тривалість розробки;

$N_{\text{год}}$ – витрати робочого часу, год;

$$t_{\text{розробки}} = 1 \cdot 21 \cdot 40 = 840 \text{ чол/год.} \quad (5.4)$$

ОЗП визначається за формулою 5.5:

$$\text{ОЗП} = t_{\text{розробки}} \cdot N \cdot K_{KB}, \quad (5.5)$$

де $t_{\text{розробки}}$ – витрати праці у чол/год;

N – зарплата за годину;

K_{KB} – коефіцієнт кваліфікації програміста, прийнято 0,75.

ОЗП становитиме за формулою 5.6:

$$\text{ОЗП} = 840 \cdot 104,17 \cdot 0,75 = 65627 \text{ грн.} \quad (5.6)$$

Відрахування на оплату соціальних внесків вираховуються у відсотковому відношенні від суми заробітної плати [3]:

$$C_{\text{соц}} = \frac{\text{ОЗП} \cdot 22\%}{100\%}$$

$$C_{\text{соц}} = \frac{65627 \cdot 22\%}{100\%} = 14438 \text{ грн.} \quad (5.7)$$

Отримані результати за пунктами (5.6) та (5.7) підсумовуються. В результаті вони складають суму 80065 грн., яку визначено основними прямими витратами.

Накладні витрати [4] включають загальногосподарські витрати на організацію робочого процесу: витрати на комунальні послуги в офісі, амортизація будівель, зарплату штатного обслуговуючого персоналу та інше. Вони визначаються в процентах (30 – 40%) від суми прямих витрат і розраховані за форм. 5.8 у 5.9:

$$C_{\text{накл}} = \frac{(\text{ОЗП} + C_{\text{соц}}) \cdot 35\%}{100\%}; \quad (5.8)$$

$$C_{\text{накл}} = \frac{(65627 + 14438) \cdot 35\%}{100\%} = 28\,022,75 \text{ грн.} \quad (5.9)$$

Впродовж усього терміну використання технічного устаткування підприємство щорічно витрачає певні кошти, пов'язані з її експлуатацією.

Витрати на експлуатацію [4] персонального комп'ютер визначаються на етапі планування розробки програмного засобу в залежності від вартості конкретної робочої станції. До експлуатаційних витрат відносяться:

- витрати на ремонт;
- оренда приміщення;
- вартість витратних матеріалів;
- заробітна плата ремонтника;
- додаткові витрати – обслуговування приміщення (прибирання, охорона, оренда, комунальні послуги);
- амортизаційні витрати на персональний комп'ютер та базовий пакет програмного забезпечення;
- витрати на електроенергію ($C_{\text{ел}}$).

Визначаються ці витрати за формулою 5.10:

$$C_{\text{ел}} = P \cdot B \cdot T_{\text{розр}}, \quad (5.10)$$

де P – потужність комп'ютера та допоміжних споживачів електричної енергії, розрахована потужність для комп'ютера та монітору прийнята 0,5 кВт/год;

B – вартість 1 кВт/година, в Україні за поточним тарифом складає 1,68 грн;

$T_{\text{розр}}$ – час роботи з ЕВМ, прирівнюється робочому часу.

Витрати на електроенергію розраховані у форм. 5.11:

$$C_{\text{ел}} = 0,5 \cdot 1,68 \cdot 840 = 705,6 \text{ грн.} \quad (5.11)$$

Для визначення заробітної плати ремонтника ($C_{\text{рем}}$) необхідно перерахувати його середньомісячну заробітну плату в години, що витрачаються за нормою на

один персональний комп'ютер. Кількість комп'ютерів, що обслуговуються одним ремонтником приймається за 50, а його зарплата за місяць – 10000 грн. [7]. Тоді на один комп'ютер буде витрачено коштів за форм. 5.12:

$$C_{\text{рем}} = \frac{C'_{\text{рем}}}{N_{\text{КОМ}}} \cdot T_{\text{міс}}, \quad (5.12)$$

де $C'_{\text{рем}}$ – середньомісячна заробітна плата;

$N_{\text{КОМ}}$ – кількість комп'ютерів на одного ремонтника.

$T_{\text{міс}}$ – час розробки програмного продукту, міс.

Заробітна плата ремонтника ($C_{\text{рем}}$) розрахована у форм. 5.13:

$$C_{\text{рем}} = \frac{10000}{50} \cdot 4,85 = 970 \text{ грн.} \quad (5.13)$$

Крім цього необхідно врахувати витрати на комплектуючі вироби ($C_{\text{КОМ}}$) [5], на які будуть замінені несправні при ремонті. Вони приймаються за 10% від вартості комп'ютера впродовж терміну його експлуатації. Тепер перерахуємо ці витрати з урахуванням часу створення програмного засобу у форм. 5.14:

$$C_{\text{КОМ}} = B_{\text{КОМ}} \cdot \frac{N_{\text{Д}}}{N_{\text{експ}} \cdot 365} \cdot \frac{10\%}{100\%}, \quad (5.14)$$

де $B_{\text{КОМ}}$ – вартість персонального комп'ютеру;

$N_{\text{Д}}$ – тривалість розробки програмного продукту у днях;

$N_{\text{експ}}$ – термін експлуатації персонального комп'ютеру.

Прийнята вартість ПК з клавіатурою, мишею і монітором 20 000 грн [11]. Сума витрат на комплектуючі частини протягом періоду розробки програми визначається у форм. 5.15:

$$C_{\text{КОМ}} = 20000 \cdot \frac{105}{5 \cdot 365} \cdot \frac{10}{100} = 115,07 \text{ грн.} \quad (5.15)$$

Сума на витратні матеріали (C_{BM}) [5] протягом всього терміну експлуатації також становить приблизно 10% від вартості комп'ютеру. Робоча станція оцінюється у 20 000 грн., термін її експлуатації – 5 років. Витрати такі ж, що й на комплектуючі вироби:

$$C_{\text{BM}} = C_{\text{КОМ}} = 115,07 \text{ грн.} \quad (5.16)$$

Амортизаційні відрахування на персональний комп'ютер (АПК) [6] визначені з положення, що амортизаційний період в даний час дорівнює терміну зміни поколінь обчислювальних машин і приймається за 3 роки. Отже, кожні 3 роки амортизаційні витрати дорівнюють вартості такого комп'ютера. Вирахуємо частку витрат на амортизаційні відрахування в період розробки програми:

$$\text{АПК} = B_{\text{КОМ}} \cdot \frac{N_{\text{Д}}}{N_{\text{експ}} \cdot 365} = 20000 \cdot \frac{105}{3 \cdot 365} = 1917,8 \quad (5.17)$$

Для визначення суми амортизаційних відрахувань на програмне забезпечення (АПЗ) [6] необхідно враховувати час розробки проекту в межах часу дії пакету прикладних програм та операційної системи, що були застосовані. Розробка проводилася в операційній системі Windows, оновлення версій якої прийнято здійснювати кожні 5 років, а за середовище розробки для поставлених задач обрано РНР, що поширюється за моделлю щомісячної підписки, тому амортизаційні відрахування на програмне забезпечення дорівнюють його вартості в перерахунок на час розробки.

Операційна система для розробки взята Windows 10, середовище розробки програм – РНР, що оплачується щомісячно за підпискою у 1200 грн/міс.

$$АПЗ_W = 6204 \cdot \frac{105}{5 \cdot 365} = 357 \text{ грн.} \quad (5.18)$$

$$АПЗ_V = 1200 \cdot 5 = 6000 \text{ грн.} \quad (5.19)$$

Розрахунок амортизаційних відрахувань на програмне забезпечення зведений у табл. 5.2.

Таблиця 5.2 – Застосоване програмне забезпечення

Найменування програмного забезпечення	Вартість програмного забезпечення, грн	Джерело придбання	Амортизаційні відрахування, грн
Windows 10 Professional	6204	https://soft.rozetka.com.ua/microsoft_fqc_09131/p3936301/	460
PHP	6000	https://www.php.net/	6000
Всього:	12204	-	6460

Додаткові витрати ($C_{\text{дод}}$) [6]: охорона, комунальні послуги, прибирання приміщень, тощо, досить складно оцінити, тому приймаються рівними 20% від заробітної плати інженера-програміста, а саме 3500 гривень на місяць.

Оренду офісу ($C_{\text{ор}}$) з площею 13 м² візьмемо 3000 гривень на місяць [8].

Загалом експлуатаційні витрати на один персональний комп'ютер будуть складати за форм. 5.20:

$$C_{\text{експ}} = C_{\text{ел}} + C_{\text{рем}} + C_{\text{КОМ}} + C_{\text{ВМ}} + АПК + АПЗ + C_{\text{дод}} + C_{\text{ор}} =$$

$$= 705,6 + 970 + 2 * 115,07 + 1917,8 + 6460 + 17500 + 15000 = 42783,54 \text{ грн} \quad (5.20)$$

Результати розрахунків приведено у табл. 5.3.

Таблиця 5.3 – Експлуатаційні витрати на ПК і ПЗ.

Найменування витрат	Витрати, грн
Витрати на електроенергію	705,6
Вартість витратних матеріалів	230,1
Витрати на ремонт	970
Амортизація персонального комп'ютера	1917,8
Амортизація програмного забезпечення	6460
Оренда приміщення	15000
Додаткові витрати	17500
Всього	42783,5

Підсумкові витрати на створення програмного продукту [6] вираховуються за форм. 5.21 у 5.22:

$$C_{\text{розробки}} = \text{ОЗП} + C_{\text{соц}} + C_{\text{накл}} + C_{\text{експ}}; \quad (5.21)$$

$$C_{\text{розробки}} = 65627 + 14438 + 28022,7 + 42783,5 = 150871,2 \quad (5.22)$$

Розрахунок витрат зведено у табл. 5.4.

Таблиця 5.4 – Кошторис витрат на розробку програмного засобу «SchoolMaster»

Найменування витрат	Витрати, грн
Основна заробітна плата	65 627
Відрахування на соціальні потреби	14 438
Накладні витрати	28 022,7
Експлуатаційні витрати	42 783,5
Всього	150871,2

За розрахованими значеннями техніко-економічних показників проекту складено кошторис витрат на розробку актуального програмного забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань «SchoolMaster». За результатами розрахунків встановлена вартість розробки програмного засобу складає 150 871,29 грн.

6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Стадії та етапи розробки програмного продукту представлені у табл. 6.1.

Таблиця 6.1 – Стадії та етапи розробки

Стадії розробки	Етапи розробки	Терміни виконання
1. Технічне завдання (ТЗ)	Функціональне та експлуатаційне призначення	15.06.21 – 30.06.21
	Вхідні та вихідні дані	02.07.21 – 10.07.21
	Вимоги до програмної документації	11.07.21 – 20.07.21
	Техніко-економічні показники	01.08.21 – 09.08.21
	Узгодження та затвердження ТЗ	01.09.21 – 15.09.21
2. Робочий проект	Розробка та програмування функціоналу програми	05.10.21 – 14.11.21
	Розробка і реалізація інтерфейсу користувача	15.11.21 – 20.11.21
	Налагодження програми	26.11.21 – 01.12.21
	Розробка, узгодження та затвердження програмної документації відповідно до вимог ГОСТ 19.101-77	01.12.21 – 03.12.21
	Проведення випробувань і корегування програми та документації за результатами випробувань	03.12.21 – 07.12.21
3. Впровадження	Підготовка і передача програми та програмної документації замовнику	08.12.21 – 10.12.21

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Контроль здійснюється за допомогою виконання запиту на формування списку наукових надбань за темою пошуку з метою знаходження помилок в програмному продукті та його специфікації. Контроль виконання роботи забезпечується керівником розробки доц. Андрющенко В.О.

Прийом програмного продукту здійснюється уповноваженою комісією.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Модель СОСОМО. [Електронний ресурс]. Режим доступу: <https://ru.wikipedia.org/wiki/СОСОМО>. Дата звернення: 13.12.21.
2. Дані щодо середньої зарплати програміста в Україні. [Електронний ресурс]. Режим доступу: <https://www.work.ua/ru/salary-dnipro-web-программист/> Дата звернення: 13.12.21.
3. Закон України «Про збір та облік єдиного внеску на загальнообов'язкове державне соціальне страхування» із змінами від 1 липня 2021 року N 1617-IX
4. Липаев В.В. Экономика производства сложных программных продуктов. – М Синтег, 2008. 432 с
5. Благодатских В.А. и др. Экономика, разработка и использование программного обеспечения ЭВМ. – М.: Финансы и статистика, 1995. – 286 с.
6. Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И.. Управление программными проектами. Достижение оптимального качества при минимуме затрат. М.: Издательский дом “Вильямс”, 2004. – 1125 с.
7. Дані щодо середньої зарплати програміста в Україні. [Електронний ресурс]. Режим доступу: <https://www.work.ua/ru/salary-компьютерный+мастер/> Дата звернення: 13.12.21.
8. Дані щодо оренди офісу в м. Дніпро. [Електронний ресурс]. Режим доступу: <https://www.olx.ua/d/obyavlenie/sdam-ofis-13i20m-mozhno-obedit-arenda-kabineta-pomeschenie-salon-tsent-IDNpGGQ.html#5d8a44c1ac;promoted> Дата звернення: 13.12.21.
9. Ціна операційної системи Windows 10. [Електронний ресурс]. Режим доступу: https://soft.rozetka.com.ua/microsoft_fqc_09131/p3936301/ Дата звернення: 13.12.21.
10. Ціна на PHP. [Електронний ресурс]. Режим доступу: <https://www.php.net/> Дата звернення: 13.12.21.
11. ПК моноблок. [Електронний ресурс]. Режим доступу: <https://ek.ua/ASUS-F5401WUAK-BA008M.htm> Дата звернення: 13.12.21.

12. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю. М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. – 38 с.
13. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2.007-98

...університету
залізничного транспорту
імені академіка В. Лазаряна
Борис БОНДАР

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДОСЛІДЖЕНЬ В ГАЛУЗІ
КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ**

Робочий проект
ЛИСТ ЗАТВЕРДЖЕННЯ
1116130.01212-01-ЛЗ

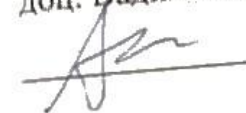
Завідувач кафедри КІТ

доц. Вадим ГОРЯЧКІН



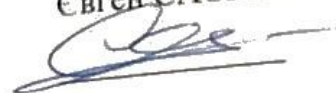
Керівник розробки

доц. Вадим АНДРІЮЩЕНКО



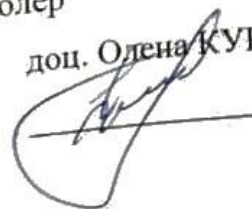
Виконавець

студент групи ПЗ2021
Євген САЄНКО



Нормоконтролер

доц. Олена КУРОП'ЯТНИК



2021

ЗАТВЕРДЖЕНО

1116130.01212-01-ЛЗ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДОСЛІДЖЕНЬ В ГАЛУЗІ
КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

Специфікація

1116130.01212-01

Аркушів 2

Позначення	Найменування	Примітка
	Документація	
1116130.01212-01-ЛЗ	Лист затвердження	
1116130.01212-01 12 01-ЛЗ	Лист затвердження	
1116130.01212-01 12 01	Текст програми	
1116130.01212-01 13 01-ЛЗ	Лист затвердження	
1116130.01212-01 13 01	Опис програми	
1116130.01212-01 31 01-ЛЗ	Лист затвердження	
1116130.01212-01 31 01	Опис застосування	
1116130.01212-01 ІЗ 01-ЛЗ	Лист затвердження	
1116130.01212-01 ІЗ 01	Керівництво користувача. Керівництво з розрахунку ARI	

ЗАТВЕРДЖЕНО
1116130.01212-01 13 01

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДОСЛІДЖЕНЬ В ГАЛУЗІ
КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ**

Опис програми

1116130.01212-01 13 01

Аркушів 18

АНОТАЦІЯ

Документ 1116130.01212-01 13 01 «Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. Опис програми» входить до складу програмної документації до програми, яка реалізує програмне забезпечення «SchoolMaster» .

У даному документі представлений опис програми: функціональне призначення, опис логічної структури, використані технічні засоби, виклик і завантаження, вхідні і вихідні дані, порядок роботи з програмою.

ЗМІСТ

1 Загальні відомості	4
2 Функціональне призначення	5
3 Опис логічної структури	6
3.1 Алгоритм програми	6
3.2 Використані методи	9
4 Використані технічні засоби	10
5 Виклик і завантаження	11
6 Вхідні дані	12
7 Вихідні дані	13
8 Опис призначеного для користувача інтерфейсу	14
9 Порядок роботи з програмою	16
10 Повідомлення	17

1 ЗАГАЛЬНІ ВІДОМОСТІ

Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань «SchoolMaster» здійснює формування списку наукових надбань за критеріями пошуку та аналіз отриманого списку у вигляді графіку кількості публікацій у часі. Програма надає користувачу інструмент для дослідження в галузі комп'ютерних наук на основі бібліографічних посилань.

Програма функціонує в середовищі операційної системи Windows починаючи з версії 7 та вище. Програма розроблена на мові PHP.

2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Програмний продукт призначений для формування списку наукових надбань за критеріями пошуку та аналіз отриманого списку у вигляді графіку кількості публікацій у часі. Його задача найбільш ефективно здійснити аналіз досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. Програма розробляється з метою організації сучасних наукових робіт.

Функціональні обмеження накладаються на обсяги одночасного запиту, що за своїм об'ємом не може перевищувати 3 ГБ.

Сферою застосування розробленого програмного продукту можуть бути всі видавничі та наукові галузі, де потрібна попередня обробка запитів на наукові роботи, дослідницькі відділи з аналізу наукових робіт.

3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ

3.1 Проектування за допомогою UML-діаграм

Для демонстрації можливих варіантів використання програми була розроблена діаграма прецедентів, що вважається першим етапом при проектуванні системи, бо наглядно показує можливі шляхи управління в програмі зі сторони користувача.

Діаграма відображає різноманітні сценарії, що можуть проходити між акторами (рядовими користувачами) та прецедентами (варіантами використання), описує функціональні можливості продукту. Перевагою даної діаграми є її наочність навіть для осіб, що безпосередньо задіяні у сфері розробки чи предметній області, на яку націлений продукт.

Для детальнішого опису системи у мові UML є наявні наступні види відносин:

- асоціація – специфікує семантичні особливості взаємодії акторів і варіантів використання в графічній моделі системи;
- узагальнення – служить для вказівки того факту, що деякий варіант використання А може бути узагальнений до варіанту використання В;
- включення – вказує, що деяка задана поведінка для одного варіанта використання включається як складовий компонента в послідовність поведінки іншого варіанту використання.
- розширення – відзначає той факт, що один з варіантів використання може приєднувати до своєї поведінки деяку додаткову поведінку, певну для іншого варіанту.

В процесі моделювання на стадії створення проектного плану було створено діаграму прецедентів (рис. 3.1). Вона містить два актора-користувача і надає наступні варіанти використання таблиця 3.1, 3.2, 3.3.

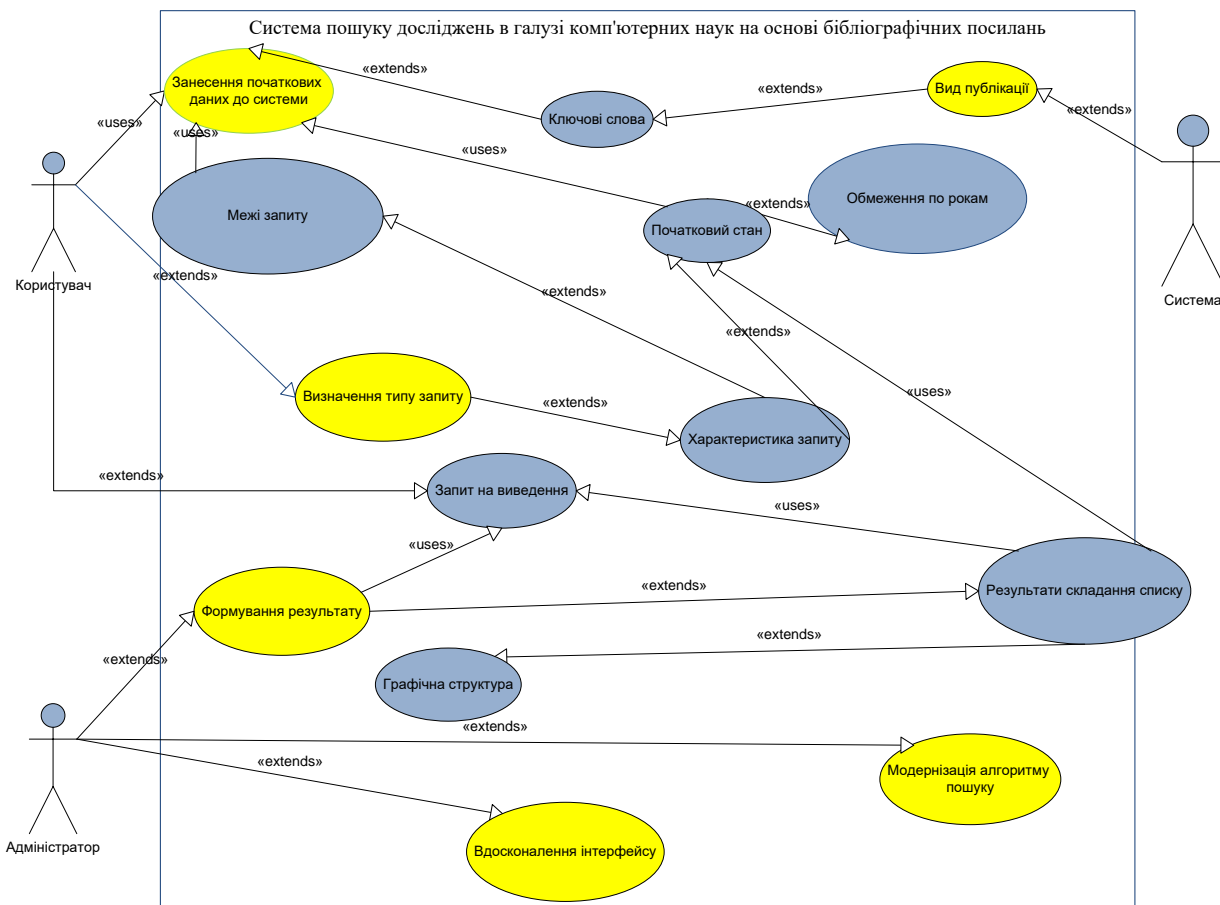


Рисунок 3.1 – Діаграма прецедентів

Таблиця 3.1 – Сценарій використання комп'ютер оператора – система

Діючі особи	Комп'ютер оператора, Система
Мета	Додати ключові слова
Передумова	Для оператора відсутні «незакриті» записи в базі даних (всі записи мають час початку і закінчення)
Успішний сценарій:	
1. Оператор включає комп'ютер. 2. Програма клієнт відправляє інформацію про наявність наукових публікацій на сервер. 3. Сервер робить запис в базу даних про наявність наукових публікацій. 4. Сервер посилає відповідь про успішний запис інформації в базу даних програмі клієнту.	
Результат	На екрані з'являється список наукових публікацій

Таблиця 3.2 – Сценарій використання оператор – система

Діючі особи	Оператор, Система
Цілі	Переглянути дані журналу наукових публікацій
Успішний сценарій:	
1. Оператор через web-інтерфейс робить запит до системи 2. Система, по запитуванім параметрам робить запит до бази. 3. Система видає оператору відповідь з інформацією про наявність наукових публікацій.	
Результат	Оператор бачить деталізацію наукових публікацій.
Розширення:	
* А	Оператор хоче внести зміни в обмеження пошуку наукових публікацій. Оператор вносить інформацію в запит через web-інтерфейс. Результат: інформація збережена.

Основна функція адміністратора системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань – підтримка актуальної інформації в пошуковому запиті. Він повинен мати інтерфейс для здійснення CRUD-операцій [5]. Сценарій для даного варіанту наведено в таблиці 3.3.

Таблиця 3.3 – Сценарій використання адміністратор – система

Діючі особи	Адміністратор, Система
Мета	Редагування сутностей бази даних
Передумова	Сервер бази даних і сервер додатка запущені і мають необхідне з'єднання
Успішний сценарій:	
1.Адміністратор через web-інтерфейс заходить на сторінку редагування даних. 2. Робиться запит на редагування даних до сервера. 3. Сервер обробляє запит, роблячи необхідні зміни в базі даних. 4. Сервер повертає відповідь з актуальною інформацією з БД	
Результат	У базі даних зберігаються зміни, зроблені адміністратором.

Діаграма активності являється ще одним з інструментів UML для відображення процесів у системі через взаємодію її функціональних модулів. Для створеної програми вона допоможе продемонструвати взаємодію окремих взятих об'єктів діяльності та подій, що поєднуються одне з одним потоками між вузловими точками (входами і виходами з елементів). Таким чином можна просто описати можливі шляхи використання програми користувачем в єдиному місці.

На рис. 3.2 представлено діаграму активності системи.

Діаграма послідовності – вид діаграми UML, що демонструє взаємодії між об'єктами у часі. На ній об'єкти можуть обмінюватися повідомленнями для послідовної взаємодії між собою. Всі процеси впорядковуються у вигляді вертикальних ліній. Надіслані повідомлення зображуються горизонтальними лініями та з'єднують 2 об'єкти, що взаємодіють. На рис. 3.3 наведено діаграму послідовностей системи (sequence diagram).

Діаграма послідовностей системи – графічна модель, яка для певного сценарію варіанту використання показує динаміку взаємодії об'єктів у часі [1, 8].

Для побудови діаграми послідовностей системи необхідно:

- ідентифікувати кожен дійову особу (об'єкт) і зобразити для неї лінію життя. Крайнім зліва на діаграмі зображується об'єкт, який є ініціатором взаємодії. Правіше зображується інший об'єкт, який безпосередньо взаємодіє з першим, і т. д.;
- з опису варіанта використання визначити безліч системних подій і їх послідовність;
- зобразити системні події у вигляді ліній із стрілкою на кінці між лініями життя дійових осіб і системи, а також вказати імена подій і списки значень, що передаються.

Діаграма компонент додатку представлена на рисунку 3.4.

Діаграма компонент – статична структурна діаграма, показує розбиття програмної системи на структурні компоненти та зв'язки (залежності) між

компонентами. У якості фізичних компонентів можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети [16].

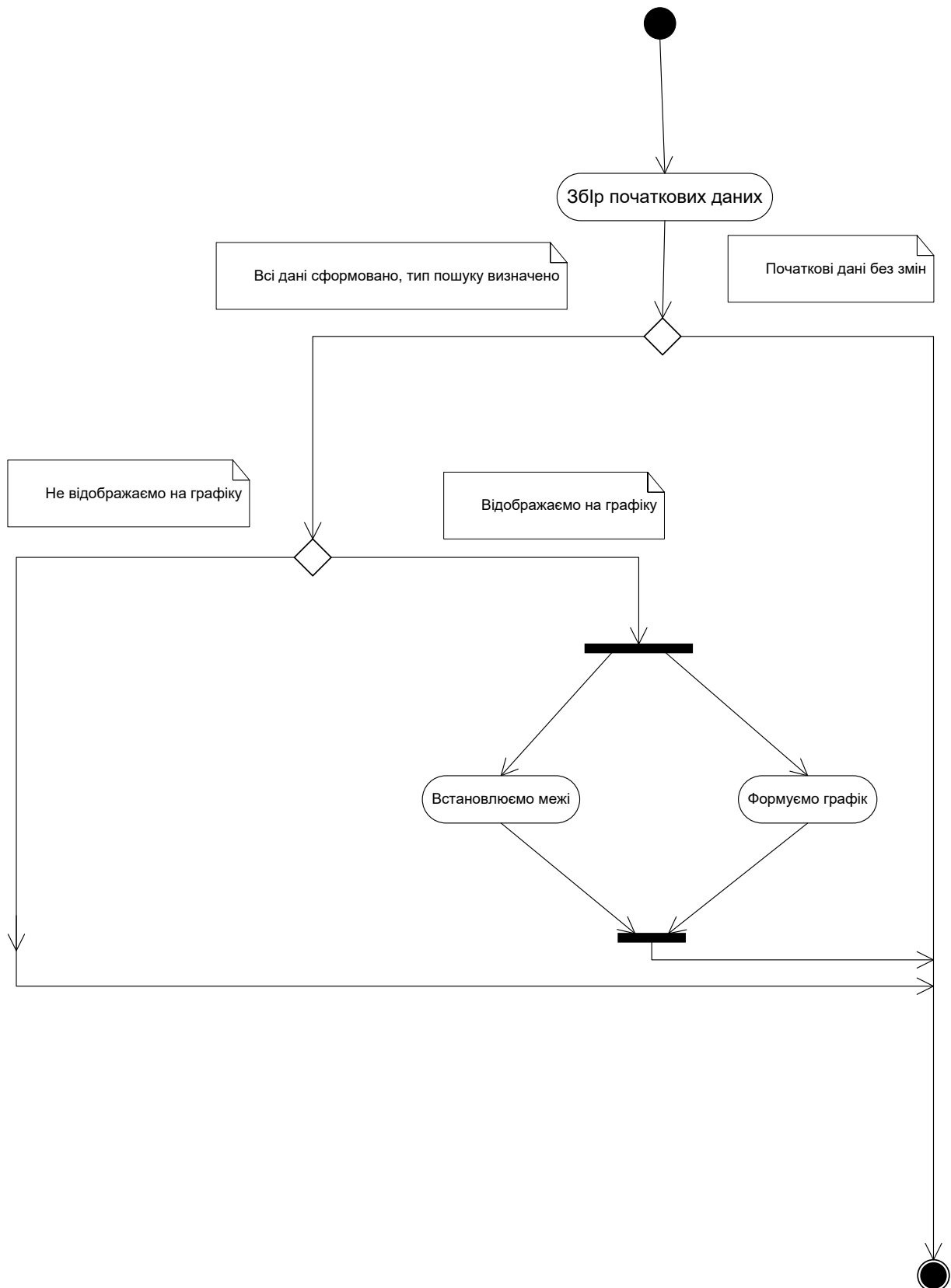


Рисунок 3.2 – Діаграма активності

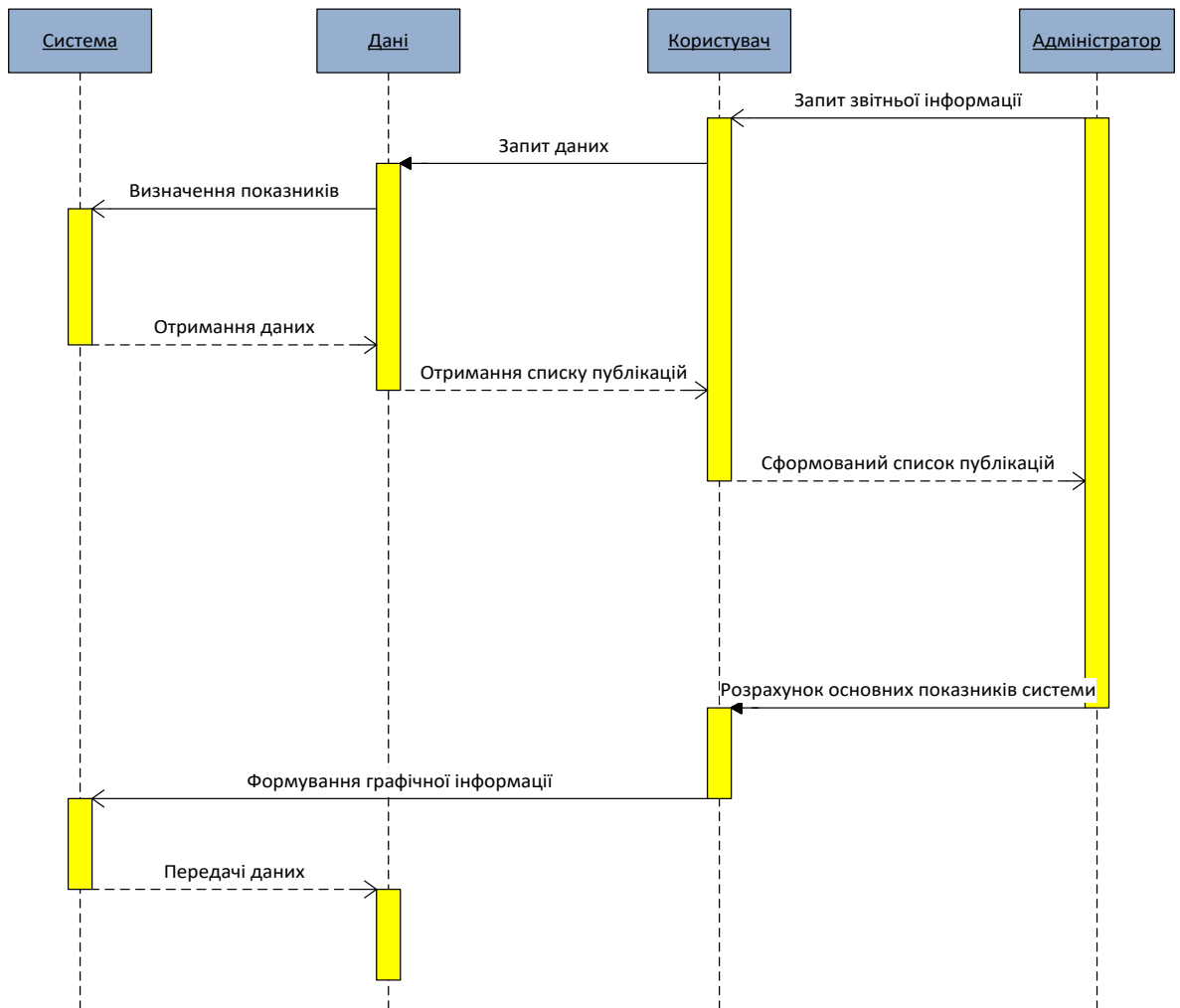


Рисунок 3.3 – Діаграма послідовності

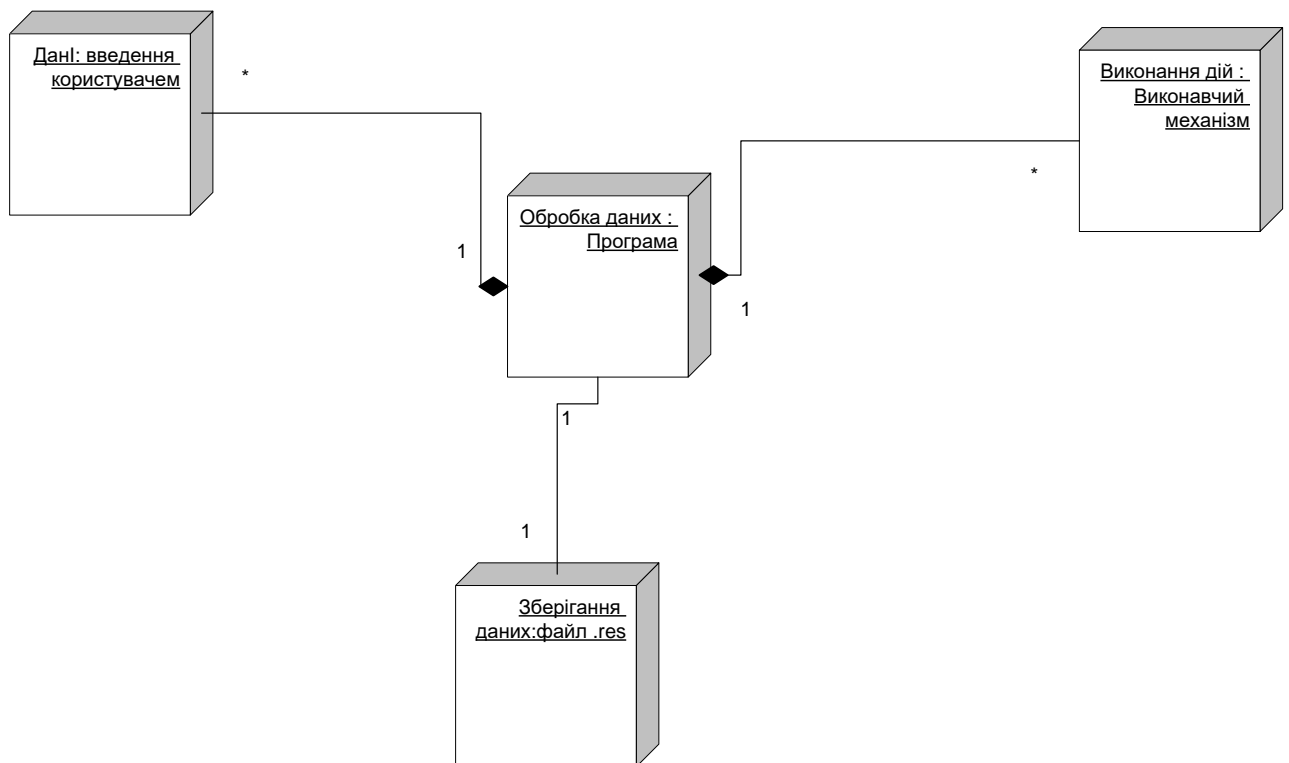


Рисунок 3.4 – Діаграма компонент

На даній діаграмі показано склад фізичних ресурсів системи (що позначаються як вузли системи) та відношень між ними.

3.2 Використані методи

Програмна система використовує методи часового оцінювання операцій вставки та пошуку Stopwatch(), що отримує часові характеристики з точністю до наносекунд, моніторингу зайнятого в оперативній пам'яті місця через діагностичні засоби.

«Клієнт – сервер» (англ. Client-server) – обчислювальна або мережева архітектура, в якій завдання або мережеве навантаження розподілені між постачальниками послуг, званими серверами, і замовниками послуг, званими клієнтами. Фактично клієнт і сервер – це програмне забезпечення.

Архітектура програми часто створюється для ілюстрації загальних властивостей програмного забезпечення (наприклад, код програми та платформа) і обладнання (наприклад, клієнт, сервер і мережеві пристрої). У літературі виділяють ряд базових шаблонів.

Архітектура додатків, як правило, проектується в термінах архітектури «клієнт-сервер», при якій один або кілька клієнтських пристроїв подають запит на інформацію з сервера. Сервер зазвичай володіє необхідною інформацією.



Рисунок 3.5 – Двошарова архітектура



Рисунок 3.6 – Тришарова архітектура

Розглянемо такі поняття клієнт-серверної архітектури: шари, рівні і зв'язок між шарами і рівнями.

Функціональність коду по всьому додатку необов'язково рівномірна. Деякі розділи коду програми краще підходять для обробки користувацького інтерфейсу, в той час як інші розділи розроблені для управління бізнес-логікою або з'єднання з базою даних або серверних систем.

Розшарування описує поділ робіт всередині коду програми на одній машині. Найчастіше шари – це не більше ніж програмні модулі, які розміщені в різних папках або каталогах на стороні клієнта або сервера.

З боку клієнта зазвичай є від нуля до трьох шарів в коді програми. З боку сервера – від одного до трьох шарів коду програми. Частково це важливо для гарного проектування програмного забезпечення, яке гарантує повторне використання коду, частково це важливо для безпеки, частково з міркувань зручності.

Клієнт з нульовою кількістю шарів коду, по суті, не має спеціального коду програми. Цей тип клієнта зазвичай згадується як тонкий клієнт і можливий в архітектурі клієнт-сервер, при якому сервер містить весь призначений для користувача код програми. Клієнт з одним до трьох шарів коду програми зазвичай називається товстим клієнтом.

Сервер, також може містити від одного до трьох шарів спеціального коду програми. Проте, за визначенням, не може бути нульової кількості шарів коду на сервері.

Шар, код якого найтісніше взаємодіє з користувачем, часто згадується в літературі як шар уявлення. Другий шар зазвичай обробляє бізнес-логіку коду. Третій шар називають рівнем доступу до даних (зв'язок з базою даних або з джерелом даних).

Цілком можлива наявність більш ніж трьох рівнів на стороні клієнта або сервера, але занадто багато шарів може привести до труднощів при управлінні.

4 ВИКОРИСТАНІ ТЕХНІЧНІ ЗАСОБИ

Технічні засоби, які використовуються при роботі програми:

- ПК, під управлінням 64-розрядної ОС Windows;
- процесор з тактовою частотою не нижче 2 ГГц;
- оперативна пам'ять не менше 6 ГБ;
- простір на жорсткому диску не менше 1 ГБ;
- маніпулятор миша;
- клавіатура;
- наявність CD/DVD приводу або USB роз'єму або LAN-адаптеру для завантаження необхідного ПЗ в ОС;
- монітор з роздільною здатністю екрану 1024x768 та більше, з підтримкою не менш ніж 256 кольорів.

5 ВИКЛИК І ЗАВАНТАЖЕННЯ

Перед використанням програми необхідно виконати наступні дії:

- виконайте запуск локального серверу OpenServer та дочекайтеся завершення запуску;
- створіть папку в кореневій папці локального серверу OpenServer та помістіть в неї всі необхідні документи;
- виконайте запуск програми з папки з сайтами.

Після запуску програма повністю готова до аналізу бібліографічних посилань.

6 ВХІДНІ ДАНІ

Вхідними даними програми, що розробляється, є:

- ключові слова;
- період публікації.

7 ВИХІДНІ ДАНІ

Вихідними даними програми, що розробляється, є:

- сформований список бібліографічних посилань складений за ключовим словом;
- графік залежності кількості посилань від року.

8 ОПИС ПРИЗНАЧЕНОГО ДЛЯ КОРИСТУВАЧА ІНТЕРФЕЙСУ

Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань «SchoolMaster» здійснює формування списку наукових надбань за ключовим словом пошуку та аналіз отриманого списку у вигляді графіку кількості публікацій у часі, інтерфейс програми зображено на рисунку 8.1

Дипломная работа'."/>

Не захищено | scholar-master:8080

Ключевые слова: [Начать поиск](#)

Год Сортировать по Страница

Данный сайт позволяет осуществить анализ динамики развития исследований в области компьютерных наук на основе библиографических ссылок.

© 2021 [Дипломная работа](#)


Рисунок 8.1 – Структура головного вікна програми

Головна сторінка програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань умовно розділена на два поля. Перше поле являє собою поле введення запиту, а друга частина це безпосередньо поле виведення результату запиту.

9 ПОРЯДОК РОБОТИ З ПРОГРАМОЮ

Перед початком роботи користувач повинен впевнитись в безпечності користуванням ПК або ноутбуком.

Для проведення тестування розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань обрано локальний сервер OpenServer.

Після запуску локального серверу, , переходимо у вкладку з сайтами (рис. 9.1) та натискаємо на назві нашого сайту (рис. 9.2) на екрані браузерa з'явиться головна сторінка (рис. 9.3).

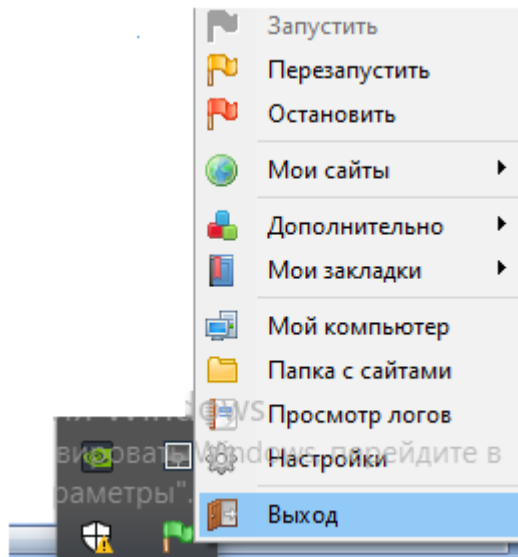


Рисунок 9.1 – Вкладення локального серверу

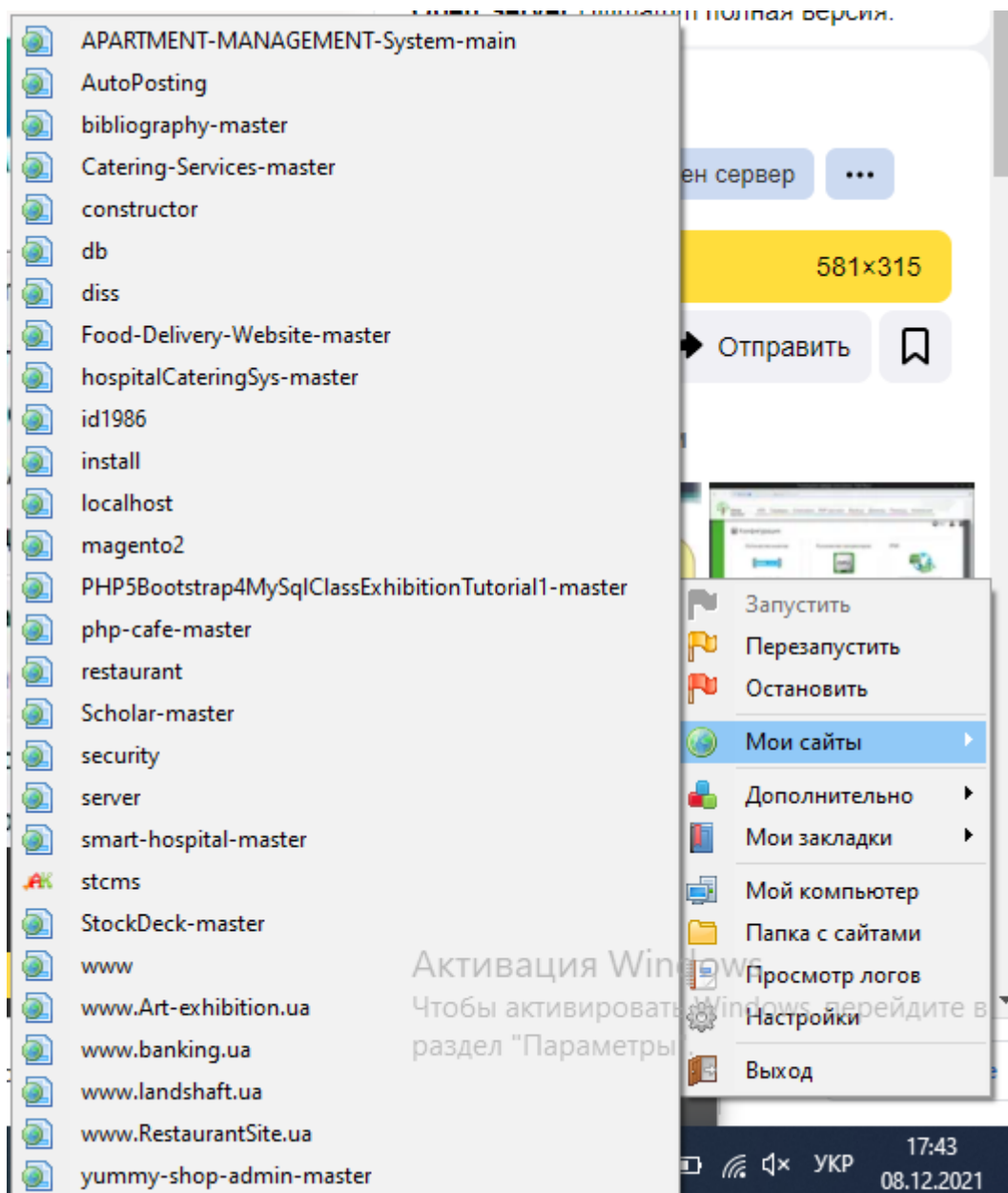


Рисунок 9.2 – Перехід по назві сайту

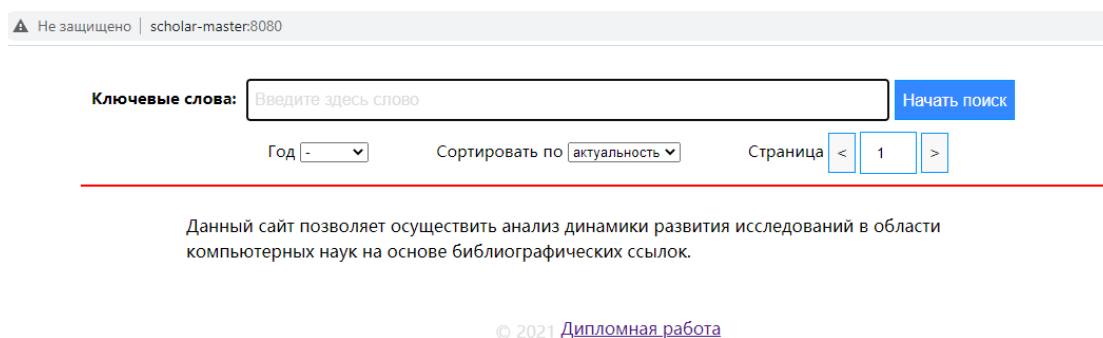


Рисунок 9.3 – Головна сторінка

Тепер протестуємо програмний додаток для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань ввівши у поле запиту ключове слово, наприклад «База даних» (рис. 9.4). Натискаємо на активній кнопці «Почати пошук» програма запропонує нам почекати поки формується загальний список публікацій (рис. 9.5) та як результат запити та очікування отримуємо результат пошуку (рис. 9.6).

ащищено | scholar-master:8080

Ключевые слова:

Год Сортировать по Страница

Данный сайт позволяет осуществить анализ динамики развития исследований в области компьютерных наук на основе библиографических ссылок.

© 2021 [Дипломная работа](#)

Рисунок 9.4 – Запит на пошук публікацій

ащищено | scholar-master:8080

Ключевые слова:

Год Сортировать по Страница

Идет загрузка ... Подождите 5 сек.

© 2021 [Дипломная работа](#)

Рисунок 9.5 – Очікування результату

Не захищено | scholar-master:8080

Ключевые слова:

База даних

Начать поиск

Год

Сортировать по

актуальность

Страница

<

1

>

<https://scholar.google.com/scholar?ie=UTF-8&q=%D0%91%D0%B0%D0%B7%D0%B0%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85>

Примерное количество результатов: 193 000 (0,06 сек.)

Мій

диапазон	<p>Реляційні бази даних як сучасний стандарт накоплення інформації в комп'ютерній системі бухгалтерського обліку [PDF] wunu.edu.ua</p> <p>ОВ Адамик , КБ Адамик - 2018 - dspace.wunu.edu.ua</p> <p>Реляційні бази є стандартом при побудові інформаційних систем підприємства и комп'ютерних систем бухгалтерського обліку, зокрема. Цьому сприяли такі їх переваги: 1) відсутність дублювання інформації; 2) при зміні значення поля, достатно змінити...</p> <p>☆ Зберегти 99 Послатися Цитовано в 13 джерелах Пов'язані статті Кількість версій: 6 99</p>
дповідн. атою	<p>База даних як об'єкт інформаційних правовідносин [PDF] univd.edu.ua</p> <p>МЮ фіменко - 2015 - dspace.univd.edu.ua</p> <p>На підставі аналізу дефініцій « база даних », «об'єкти інформаційних правовідносин» доводиться, для базы даних може розглядатися як об'єкт інформаційних правовідносин. Наголошено, надає доступ к базам данных на межсайтовых информационных правовідносин потребу...</p> <p>☆ Зберегти 99 Послатися Цитовано в 5 джерелах Пов'язані статті Кількість версій: 2 99</p>
патенти	<p>Бази і сховища даних –інформаційний фундамент бухгалтерського обліку та аналізу [PDF] wunu.edu.ua</p>

Рисунок 9.6 – Результат роботи розробленого програмного додатку для системи пошуку

Програмний додатком передбачено можливість виведення результату за роком, для цього необхідно із випадваючого списку (рис. 9.7) обрати необхідний рік, наприклад обираємо 2021 рік (рис. 9.8) програма просить знову почекати поки відбувається вибірка наукових публікацій за вказаним роком, результат отримуємо по року, який обрано (рис. 9.9).

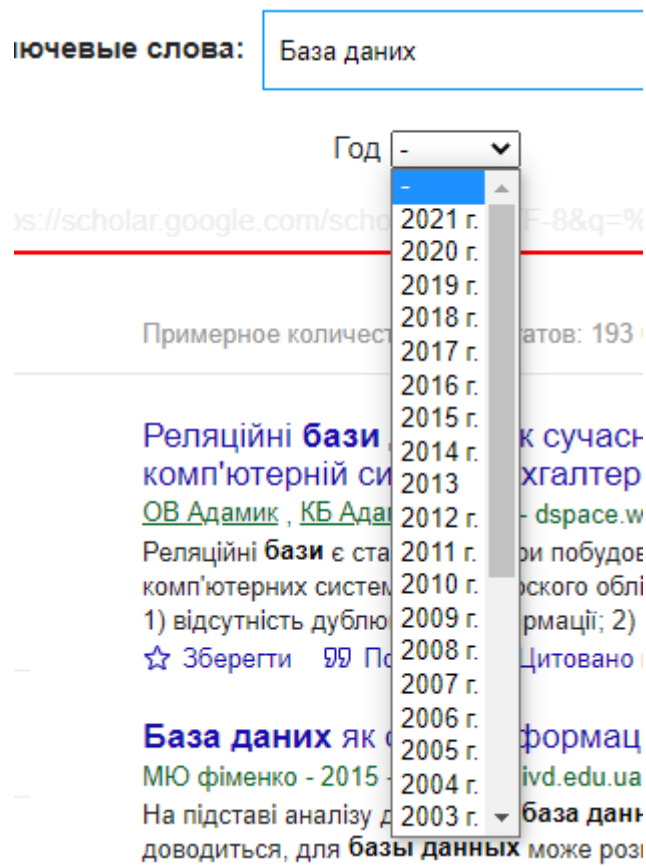


Рисунок 9.7 – Результат роботи розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

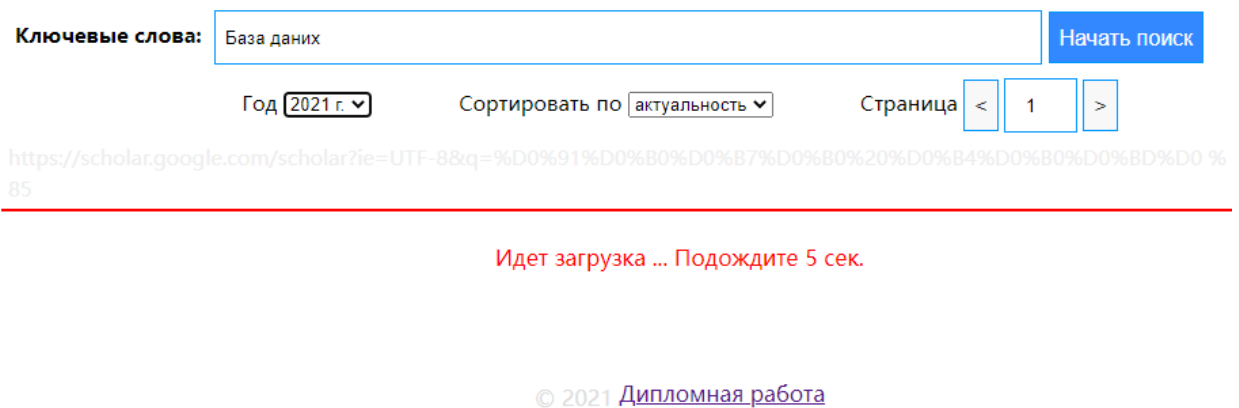


Рисунок 9.8 – Очікування результату сортування за роком

Також розроблений програмний продукт дозволяє здійснювати аналіз динаміки розвитку досліджень в окресленій галузі (рис. 9.10). Графічне представлення результатів вибірки у певній науковій сфері досліджень дозволяє об'єктивно оцінювати сучасні наукові набутки за сферою знань.



Рисунок 9.10 – Результат роботи розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

10 ПОВІДОМЛЕННЯ

В табл. 10.1 представлені повідомлення користувачу, що можуть з'явитися у процесі роботи програми.

Таблиця 10.1 – Повідомлення, що з'являються в процесі роботи програми

Текст повідомлення	Кому призначене	Опис ситуації	Рекомендовані дії
Йде завантаження почекайте	Користувач	Для формування списку потрібен час.	Чекати.

ЗАТВЕРДЖЕНО
1116130.01212-01 ІЗ 01-ЛЗ

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДОСЛІДЖЕНЬ В ГАЛУЗІ
КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ**

Керівництво користувача. Керівництво дослідника

1116130.01212-01 ІЗ 01

Аркушів 7

АНОТАЦІЯ

Документ 1116130.01212-01 ІЗ 01 «Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. Керівництво користувача. Керівництво дослідника» входить до складу програмної документації дипломного проекту.

У даному документі представлені призначення та умови застосування, підготовка до роботи, опис операцій, аварійні ситуації та рекомендації до засвоєння.

ЗМІСТ

Вступ	4
1 Призначення та умови застосування	5
2 Підготовка до роботи	6
2.1 Склад і зміст дистрибутивного носія даних	6
2.2 Порядок завантаження даних і програм	6
2.3 Порядок перевірки працездатності	6
3 Опис операції	7
3.1 Функція пошуку	7
3.2 Функція графіку	6
4. Аварійні ситуації	7

ВСТУП

Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань «SchoolMaster» здійснює формування списку наукових надбань за ключовим словом пошуку та аналіз отриманого списку у вигляді графіку кількості публікацій у часі.

Для використання програми, користувачу необхідно вміти працювати в операційній системі Windows, мати навички користування мишею і клавіатурою, розуміти принцип роботи локального серверу.

Перед роботою необхідно ознайомитись з наступними документами:

- опис програми;
- керівництво користувача.

1 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань «SchoolMaster» здійснює формування списку наукових надбань за ключовим словом пошуку та аналіз отриманого списку у вигляді графіку кількості публікацій у часі.

Для виконання програми необхідна наявність ПК або ноутбуку, на якому встановлена Windows 7 або новіша версія. Конфігурація ПК для функціонування програмного продукту повинна бути наступна:

- ПК, під управлінням 64-розрядної ОС Windows;
- процесор з тактовою частотою не нижче 2 ГГц;
- оперативна пам'ять не менше 6 ГБ;
- простір на жорсткому диску не менше 1 ГБ;
- маніпулятор миша;
- клавіатура;
- наявність CD/DVD приводу, USB роз'єму або LAN-адаптеру для завантаження необхідного ПЗ в ОС;
- монітор з роздільною здатністю екрану 1024x768 та більше, з підтримкою не менш ніж 256 кольорів.

Для користування програмою користувач повинен вміти працювати в операційній системі сімейства Windows та локальним сервером.

2 ПІДГОТОВКА ДО РОБОТИ

2.1 Склад і зміст дистрибутивного носія даних

Дистрибутив програмного продукту містить у собі файли програми.

2.2 Порядок завантаження даних і програм

Для проведення тестування розробленого програмного додатку для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань обрано локальний сервер OpenServer. Папка з розробленим програмним додатком для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань лежить у кореневій папці сервера. Саме з відси і відбувається запуск розробленого програмного продукту.

Після запуску локального серверу, переходимо у вкладку з сайтами та натискаємо на назві нашого сайту на екрані браузерa з'явиться головна сторінка.

2.3 Порядок перевірки працездатності

Тепер протестуємо програмний додаток для системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань ввівши у поле запити ключове слово, наприклад «База даних». Натискаємо на активній кнопці «Почати пошук» програма запропонує нам почекати поки формується загальний список публікацій та як результат запити та очікування отримуємо результат пошуку. Програмний додатком передбачено можливість виведення результату за роком, для цього необхідно із випадуючого списку обрати необхідний рік, наприклад обираємо 2021 рік, програма просить знову почекати поки відбувається вибірка наукових публікацій за вказаним роком, результат отримуємо по року, який обрано.

3 ОПИС ОПЕРАЦІЇ

3.1 Функція пошуку

Виконуйте складання списку бібліографічних посилань за ключовим словом.

3.3 Функція формування графіку

Будуйте графік залежності кількості публікацій від року.

4. АВАРІЙНІ СИТУАЦІЇ

При повторюваних відмовах технічних засобів, які унеможливають роботу з програмою, необхідно звернутись до системного адміністратора.

У випадку одиничного збою треба виконати перезавантаження програмного засобу або системи. При повторних збоях звертатися до системного адміністратора.

У разі виявлення помилок у даних необхідно пересвідчитися у правильності алгоритму користування програмою, перевірити вхідні дані, після чого перезапустити програму або систему, якщо проблема була підтверджена.

У разі виявлення некоректного виводу зображення на електронний вимкнути персональний комп'ютер та від'єднати його та периферійні пристрої від мережі, перевірити контакт кабелю дисплею від комп'ютеру до монітору. Якщо проблема не була усунута, викликати адміністратора.

У разі відмови магнітних або твердотільних носіїв треба встановити програму з диску на справний магнітний або твердотільних носій.

У випадках виявлення несанкціонованого втручання в дані одразу повідомити про це системного адміністратора.

В інших аварійних ситуаціях повідомити системного адміністратора про аварійну ситуацію.

ЗАТВЕРДЖЕНО

1116130.01212-01 12 01-ЛЗ

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДОСЛІДЖЕНЬ В ГАЛУЗІ
КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ**

Текст програми

1116130.01212-01 12 01

Аркушів 29

АНОТАЦІЯ

Документ 1116130.01212-01 12 01 «Програмне забезпечення для аналізу досліджень в галузі комп'ютерних наук на основі бібліографічних посилань. Текст програми» входить до складу програмної документації дипломного проекту.

У даному документі представлено схему програми та текст.

ЗМІСТ

1 Структура програми	4
2 Текст програми	6

Відповідно до розглянутих раніше концептуальних вимог до системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань, найбільш підходящою архітектурою для розробки програми є архітектура клієнт-сервер. Використання даного архітектурного підходу дасть ряд переваг [6]:

- Загальна серверна логіка при обробці запитів
- Загальна база даних, що надається в якості ресурсу,
- Незалежність клієнтів один від одного

Розглянемо загальну схему взаємодії в рамках розробленої системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань представленої на рисунку 1.

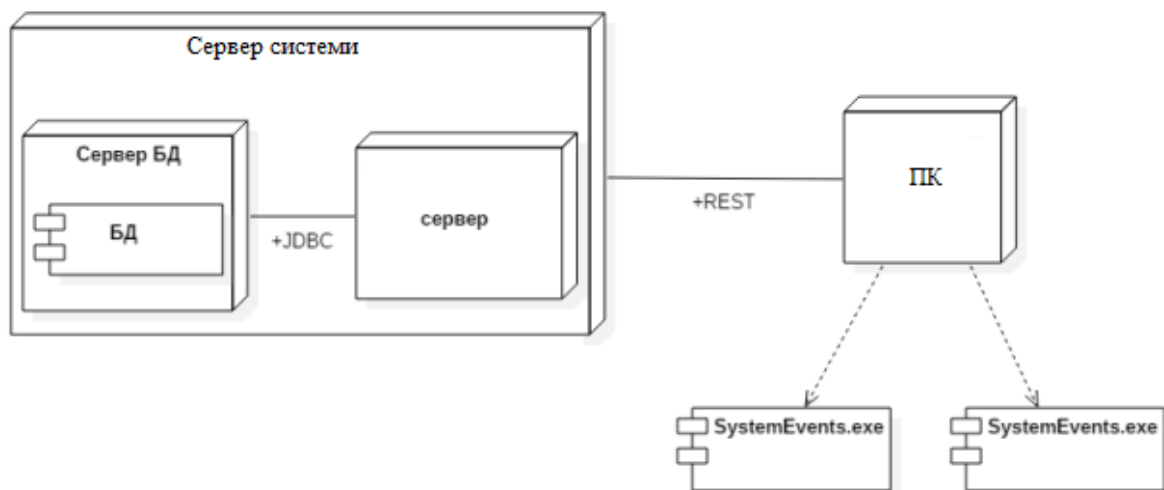


Рисунок 1 – Архітектура системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань

На рисунку 1 наведені основні структурні частини системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань:

- сервер бази даних, на якому безпосередньо знаходиться база даних і який забезпечує запити до неї з боку сервера додатку і відповіді на них.
- сервер програми – ядро розробленої системи. Містить основну бізнес логіку додатка, а також пов'язує всі інші частини інфраструктури

розробленої інформаційної системи пошуку досліджень в галузі комп'ютерних наук на основі бібліографічних посилань.

- комп'ютери-клієнти – робочі комп'ютери операторів, на кожен з яких встановлений свій екземпляр програми-клієнта, що передає інформацію на сервер.

2 ТЕКСТ ПРОГРАММ

2.1 getScholarAPI.php

```
<?php
header("Content-type: text/html; charset=utf-8");
define('DEBUG_MODE',false);//
require 'common/function.php';//

if(!isset($_POST['submit'])){
    echo ('<h1>Access Denied!</h1><h2>Jump to <a
href="index.php">Home</a> page in <span id=jumpTo
style="color:red"></span> second(s).</h2>');
    echo      "<script      type='text/javascript'
src='js/ajax.js'></script>
    <script>countDown(3, 'index.php');</script>";
    die();
}

//
//$url="http://www.google.com.hk/search?ie=UTF-8&q";
if(isset($_POST['keyword2'])){
    $keyword1=$_POST['keyword2'];
myLog($keyword1);//
    $keyword2=urlencode($keyword1);

    //
    //echo '<hr>'.$keyword1,'<hr> '.$a;    die();

    //
    $since=myIsset('since',0);
    $sortBy=myIsset('sortBy',0);
    $page=myIsset('page',1);

    //
    $page=$page<=10?$page:10;
    $page=$page>=1?$page:1;

    $url="http://scholar.google.com/scholar?";
```

```
//
if($page>1){$url .= '&start=' . ($page-1) . '0';}

//
$url .= "ie=UTF-8&q=$keyword2";

//
if($since!=0){$url .= "&as_ylo=" . $since;}
if($sortBy==1){$url .= "&scisbd=1";}
}else{
    $url="http://scholar.google.com/";
};

//
if(DEBUG_MODE){
    echo      "<pre>";    echo      "<hr>$url<hr>";
var_dump($_POST);
    echo "<hr>"; print_r($_POST); die();
}

//
$handle = fopen("$url", "rb");

$contents = "";
if(!$handle){
    die("Ошибка открытия файла ! ");
}
while (!feof($handle)) {
    $contents .= fread($handle, 8192);
}
    echo $contents;
fclose($handle);
?>
```

getPic.php

```
<p style="color:#ccc">1. загрузка файла
изображения займет не менее 5 секунд.<br>
```

```
<?php
error_reporting(E_ALL);
ini_set("display_errors","On");

require 'common/function.php';
//https://www.economist.com/sites/default/files/
images/print-edition/20190615_CNP003_1.jpg

print("2.URL format:
www.xx.com/getPic.php?url=https://xx.com/xx2.png
</p><pre>");
```

```
//1. get pic url
if(!empty($_GET['url'])){
    $file_url = $_GET['url'];
}else{
    $file_url =
"https://static01.nyt.com/images/2019/07/16/video/16vid-retro/14vid-retro-driverless-photo-
facebookJumbo.jpg";

    //"https://www.baidu.com/img/bd_logo1.png";
}
```

```
echo "3.pic_url=".$file_url."<br>";
$arr=explode("/", $file_url);
$n=count($arr)-1;
$save_to=$arr[$n];
```

```
//2. download file using file_get_contents
echo "4.Save_to <a target=_blank
href=/img/". $save_to.">$save_to</a><br>";
downloadfile($file_url, "img/". $save_to);
```

```
//3.show img
```

```
sleep(5); //sleep for download
echo "<img src=img/". $save_to." /><br>";
```

```
//4. show /img/ file list
$arr_file = array();
tree($arr_file, "./img/");
print_r($arr_file);
```

getPic2.php

```
<p style="color:#ccc">1. загрузка файла
изображения займет не менее 5 секунд.<br>
```

```
<?php
error_reporting(E_ALL);
ini_set("display_errors","On");
```

```
require 'common/function.php';
print("2.URL format:
www.xx.com/getPic2.php?url=https://xx.com/xx2.png
</p><pre>");
```

```
//1. get pic url
if(!empty($_GET['url'])){
    $file_url = $_GET['url'];
}else{
    $file_url =
"https://www.baidu.com/img/bd_logo1.png";
}
```

```
echo "3.pic_url=".$file_url."<br>";
$arr=explode("/", $file_url);
$n=count($arr)-1;
$save_to=$arr[$n];
```



```
//2. download file using file_get_contents
echo      "4.Save_to      <a      target=_blank
href=/img/". $save_to.">$save_to</a><br>";

$curl = curl_init($file_url);
curl_setopt($curl,CURLOPT_RETURNTRANSFER,1);
$imageData = curl_exec($curl);
curl_close($curl);
$tp = fopen("img/". $save_to,"a");
fwrite($tp, $imageData);
fclose($tp);

//3.show img
sleep(5); //sleep for download
echo "<img src=img/". $save_to." /><br>";

//4. show /img/ file list
$arr_file = array();
tree($arr_file, "./img/");
print_r($arr_file);
```

getPic3.php

```
<?php
header("Content-type: text/html; charset=utf-8");
error_reporting(E_ALL);
ini_set("display_errors","On");

require 'common/function.php';
//
?>

<head>
<title>1.0</title>
<link      rel="stylesheet"
href="css/scholarPage.css">
<style>
#myWrap{width:1000px; margin:35px auto; }
#mySearchLogo{
padding:5px;
```

```
margin:5px auto;
width:100%;
font-family:"Microsoft YaHei";
}
#mySearchResult input[type="text">{width:60%;
border:1px solid #0096ff;
height:40px;
padding:0 7px;
}
#mySearchResult
input[type="submit"]{width:100px; border:1px
solid #0096ff;
height:40px;
}

#gs_nml{
display:none;
}
</style>
</head>

<body>
<div id=myWrap>
<form      method="POST"      target=""
id=mySearchResult      onsubmit="return
checkForm()">
<b id="mySearchLogo">Picture URL:</b>
<input      type="text"      name="keyword"
value="<?php
if(isset($_POST['keyword2']))){
echo
ascii2str($_POST['keyword2']);
}
?>">
<input      type="submit"      name="submit"
class=btn value="Get Image">
</form>

<p style="color:#ccc">1.It will take at least
5s to download the picture file.<br>
```

2.Picture	URL	format:	
https://www.google.com/images/branding/googlel			<pre>
ogo/1x/googlelogo_color_272x92dp.png </p>			<?php
			if(isset(\$_POST['keyword2'])){
			\$url=\$_POST['keyword2'];
			if(trim(\$url)==""){
			die("Pls input the URL");
			}
			}else{
			die("Pls input the URL");
			}
			//
			\$file_url=ascii2str(\$url);
			//
			echo "3.pic_url=".\$file_url." ";
			\$arr=explode("/", \$file_url);
			\$n=count(\$arr)-1;
			\$save_to=\$arr[\$n];
			//2. download file using file_get_contents
			echo "4.Save_to <a target=_blank
			href=/img/".\$save_to.">\$save_to ";
			downloadfile(\$file_url, "img/".\$save_to);
			//3.show img
			sleep(5); //sleep for download
			echo " ";
			//4. show /img/ file list
			\$arr_file = array();
			tree(\$arr_file, "./img/");
			//
			for (\$x=0; \$x<count(\$arr_file); \$x++) {
			echo \$x." <a target=_blank
			href=img".\$arr_file[\$x].">img".\$arr_file[\$x]."
			 ";
			}
			>?

```
</div>
</body>
```

gg.php

```
<?php
header('Content-Type:application/json;
charset=utf-8'); #
#header("Content-type: text/json; charset=utf-
8");
define('DEBUG_MODE',false);//
require 'common/function.php';//

$data=file_get_contents('php://input') ; //
$data = (array)json_decode( $data );//

//echo json_encode($data); die();

if(!isset($data['submit'])){
    echo ('<h1>Access Denied!</h1><h2>Jump
to <a href="index.php">Home</a> page in <span
id=jumpTo style="color:red"></span>
second(s).</h2>');
    echo "<script type='text/javascript'
src='js/ajax.js'></script>
<script>countDown(3,
'index.php');</script>";
    die();
}

$kw=$data['keyword']; myLog($kw);
$page=$data['page'];

//
$page=$page<=10?$page:10;
$page=$page>=1?$page:1;

#$kw2=urldecode( $kw );
//
```

```
// $url="https://www.baidu.com/s?wd=".$kw;
$url="https://www.google.com/search?q=".$kw;
if($page>=2){
    $url=$url."&start=".( $page-1)*10;
}

// $url="https://cn.bing.com/search?q=".$kw;
// $url="https://www.bing.com/search?q=".$kw;

#$url="https://www.showmyip.com/";
$arr=array(
    'status'=>"ok",
    'msg'=>"a msg here;",
    'kw'=>urldecode( $data['keyword'] ),
    'page'=>$page,
    'url'=>$url,
    #'html'=>'html' <a>code
here</a>',#curl_get_contents($url),
    'html'=>curl_get_contents($url),
    #'kw'=>urlencode(
    ascii2str($data['keyword']) )
);
echo json_encode( $arr );

#
```

gs.php

```
<?php
header('Content-Type:application/json;
charset=utf-8'); #
#header("Content-type: text/json; charset=utf-
8");
define('DEBUG_MODE',false);//
require 'common/function.php';//

$data=file_get_contents('php://input') ; //
$data = (array)json_decode( $data );//

//echo json_encode($data); die();

if(!isset($data['submit'])){
```

```

        echo ('<h1>Access Denied!</h1><h2>Jump
to <a href="index.php">Home</a> page in <span
id=jumpTo          style="color:red"></span>
second(s).</h2>');

        echo "<script type='text/javascript'
src='js/ajax.js'></script>
        <script>countDown(3,
'index.php');</script>";
        die();
    }

//
//$url="http://www.google.com.hk/search?ie=UTF
-8&q";
if(isset($data['keyword'])){
    $kw=$data['keyword']; myLog($kw);//
    //$keyword2=urlencode($keyword1);
    //todo
    //echo '<hr>'.$keyword1,'<hr>'.'$a;
    die();

    //
    $since=$data['since'];
    $sortBy=$data['sortBy'];

    $page=$data['page'];
    //
    $page=$page<=10?$page:10;
    $page=$page>=1?$page:1;

    $url="https://scholar.google.com/scholar
?";

    //
    if($page>1){
        //$url .= '&start=' . ($page-1) .
'0';

        $url .= "&start=".(($page-1)*10;
    }

```

```

//
$url .= "ie=UTF-8&q=$kw";

//
if($since!=0){$url .= "&as_ylo=" .
$since;}
if($sortBy==1){$url .= "&scisbd=1";}
}else{
    $url="https://scholar.google.com/";
};

//
if(DEBUG_MODE){
    echo "<pre>"; echo "<hr>$url<hr>";
var_dump($_POST);
    echo "<hr>"; print_r($_POST); //die();
}

#$url="https://www.showmyip.com/";
$arr=array(
    'status'=>"ok",
    'msg'=>"a msg here;",
    'kw'=>urlencode( $data['keyword'] ),
    'page'=>$page,
    'url'=>$url,
    #'html'=>'html' <a>code
here</a>','#curl_get_contents($url),
    'html'=>curl_get_contents($url),
);
echo json_encode( $arr );

#

```

help.php

```

<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">

```

```

<meta                                name="viewport"
content="width=device-width, initial-scale=1">
  <!-- Приведенные выше 3 метатега * должны
* быть размещены первыми, а любой другой
контент * должен * следовать за ним. ! -->
  <title>Дипломная работа</title>

  <!-- Bootstrap -->
  <link
href="//cdn.bootcss.com/bootstrap/3.3.5/css/bo
otstrap.min.css" rel="stylesheet">

  <!-- HTML5 shim and Respond.js for IE8
support of HTML5 elements and media queries --
>
  <!-- WARNING: Respond.js doesn't work if
you view the page via file:// -->
  <!--[if lt IE 9]>
    <script
src="//cdn.bootcss.com/html5shiv/3.7.2/html5sh
iv.min.js"></script>
    <script
src="//cdn.bootcss.com/respond.js/1.4.2/respon
d.min.js"></script>
    <![endif]-->

<style>
body{padding-bottom:50px;}
.jumbotron{margin-top:60px;}
#carousel-example-generic{
    height:200px;
}
</style>
</head>
<body role="document">
<!-- Fixed navbar -->
  <nav class="navbar navbar-inverse navbar-
fixed-top">
    <div class="container">
      <div class="navbar-header">

```

```

        <button type="button" class="navbar-
toggle collapsed" data-toggle="collapse" data-
target="#navbar" aria-expanded="false" aria-
controls="navbar">
          <span class="sr-only">Переключить
навигацию</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a                                class="navbar-brand"
href="#">[Ученый]</a>
      </div>
      <div id="navbar" class="navbar-
collapse collapse">
        <ul class="nav navbar-nav">
          <li class="active"><a
href="index.php">Страница поиска</a></li>
          <li><a href="#about">关于拂晓
</a></li>
          <li><a href="#contact">联系我们
</a></li>
          <li class="dropdown">
            <a href="#" class="dropdown-
toggle" data-toggle="dropdown" role="button"
aria-haspopup="true" aria-expanded="false">搜索
选项 <span class="caret"></span></a>
            <ul class="dropdown-menu">
              <li><a href="#">关键词搜索
</a></li>
              <li><a href="#">限定时间
</a></li>
              <li><a href="#">结果排序
</a></li>
              <li role="separator"
class="divider"></li>
              <li class="dropdown-header">翻
页</li>
            </ul>
          </li>

```

```

        </ul>
    </div><!--/.nav-collapse -->
</div>
</nav>

<div class="container theme-showcase"
role="main">
    <!-- Main jumbotron for a primary
marketing message or call to action -->
    <div class="jumbotron">
        <h1>拂晓学术搜索</h1>
        <p>拂晓学术搜索:由谷歌粉丝建立的方便大家稳
定高速地使用谷歌学术搜索的搜索引擎。谷歌搜索如果您
所在地区无法打开google或者比较卡,欢迎使用拂晓搜索
进行谷歌学术搜索。</p>
    </div>

    <div id="carousel-example-generic"
class="carousel slide" data-ride="carousel">
        <ol class="carousel-indicators">
            <li data-target="#carousel-example-
generic" data-slide-to="0"
class="active"></li>
            <li data-target="#carousel-example-
generic" data-slide-to="1"></li>
            <li data-target="#carousel-example-
generic" data-slide-to="2"></li>
        </ol>

        <div class="carousel-inner"
role="listbox">
            <div class="item active">
                
            </div>

```

```

        <div class="item">
            
        </div>

        <div class="item">
            
        </div>

        <a class="left carousel-control"
href="#carousel-example-generic" role="button"
data-slide="prev">
            <span class="glyphicon glyphicon-
chevron-left" aria-hidden="true"></span>
            <span class="sr-
only">Previous</span>
        </a>

        <a class="right carousel-control"
href="#carousel-example-generic" role="button"
data-slide="next">
            <span class="glyphicon glyphicon-
chevron-right" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>
</div>

```

```
<div class="text-center">&copy; 2021
    <?php include('common/footer.php');?>
</div>

</div>
    <!-- jQuery (necessary for Bootstrap's
JavaScript plugins) -->
    <script
src="//cdn.bootcss.com/jquery/1.11.3/jquery.mi
n.js"></script>
    <!-- Include all compiled plugins (below),
or include individual files as needed -->
    <script
src="http://apps.bdimg.com/libs/bootstrap/3.3.
4/js/bootstrap.min.js"></script>
    </body>
</html>
```

index2.php

```
<!DOCTYPE html>
<html>
<head>
<title>Поиск библиографических ссылок</title>
<meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1">
<meta http-equiv="content-type"
content="text/html; charset=utf-8">
<!-- Bootstrap -->
    <link
href="//cdn.bootcss.com/bootstrap/3.3.5/css/bo
otstrap.min.css" rel="stylesheet">
<style>
body{padding:0; margin:0; font-
family:"Microsoft YaHei";}
div,input {vertical-
align:middle}/*cssТекстовое поле и кнопка
внутри не выровнены*/
```

```
#myWrap{width:1000px; margin:35px auto; }
#mySearchLogo{
padding:5px;
margin:5px auto;
width:100%;
font-family:"Microsoft YaHei";
}
#mySearchResult{margin:0 30px;}
#mySearchResult input{border:1px solid
#0096ff; height:40px;}
#mySearchResult input[type="text"]{width:70%;
}
#mySearchResult
input[type="submit"]{width:100px;}

#gs_nml{
display:none;
}
```

```
.card{
background: #0096ff;
width: 100px;
position: relative;
margin: 2em 2em;
padding: 15px;
box-shadow: 0px 3px rgba( 0, 0, 0, 0.1);
color: #E8E8E8;
font-size: 14px;
-moz-border-radius: 2px;
-webkit-border-radius: 2px;
border-radius: 2px;
}
.card.announce {
position: absolute;
margin: 2em 5px;
right: 0;
```

```

        top: 0;
    }
    .card a{
        color:#fff;
    }
    .light{
        background-color: #f8f8f8;
        color: #222;
        border: 1px solid #c6c6c6;
    }

    #erWeiMa{
        position:absolute;
        top:30px;
        padding:5px;
    }
    #erWeiMa img{width:120px; }

</style>
</head>
<body>

<div id="url_info" class='card announce'>
    <b>Диплом</b><h2>1</h2>

</div>

```

```

<div id=myWrap>

```

```

<form          method="POST"          target=""
id=mySearchResult>
    <b id="mySearchLogo">
        Поиск:
    </b>
    <input      type="text"      name="keyword"
value="<?php
        if(isset($_POST['keyword']))){
            echo $_POST['keyword'];
        }
        ?>">
    <input      type="submit"      name="submit"
value="Scholar">
</form>

```

```

<div class="container" style='margin:15px 0;'>

    <div class="jumbotron">
        <h1>Диплом</h1>
        <p>Мой диплом.</p>
        <p>
            Поиск      <a      href='index.php'
type="button"      class="btn      btn-lg      btn-
danger">лучших      научных      публикаций</a>
Поиск.</p>
    </div>

```

```

<?php
require 'common/function.php';
define('DEBUG_MODE',true);

//$url="http://www.google.com.hk/search?ie=UTF
-8&q=武汉";
if(isset($_POST['keyword'])){
    $keyword=$_POST['keyword'];

```



```

        myLog('[Basic]'.$keyword);//
        $url="http://scholar.google.com.cn/scholar?ie=UTF-8&q=$keyword";
    }else{
        $url="http://scholar.google.com.cn/";
        myLog('[Basic] visit');//
    };

    if(isset($_POST['keyword'])){
        $keyword=$_POST['keyword'];
        if(trim($keyword)==""){
            die("Пожалуйста, введите ключевое слово");
        }
    }else{
        die("Пожалуйста, введите ключевое слово");
    }
}

```

```

        $handle = fopen($url, "rb");

        $contents = '';
        if(!$handle){
            die("Ошибка открытия файла!");
        }
        while (!feof($handle)) {
            $contents .= fread($handle, 8192);
        }
        echo $contents;
        fclose($handle);
    ?>

```

</div>

```

<style>
#gs_hdr,#gs_ab_rt,#gs_gb,#gs_lnv,#gs_n,#gs_ftr
,
#gs_hp_main #gs_hp_tsi {display:none;}

#gs_ccl { margin-left: 10px;}

```

</style>

```

<div style='text-align:center;'
role="contentinfo">
    <script type="text/javascript">var
cnzz_protocol = (("https:" ==
document.location.protocol) ? " https://" : "
http://");document.write(unescape("%3Cspan
id='cnzz_stat_icon_1256189537'%3E%3C/span%3E%3
Cscript src='\" + cnzz_protocol +
"s95.cnzz.com/stat.php%3Fid%3D1256189537%26onl
ine%3D1%26show%3Dline'
type='text/javascript'%3E%3C/script%3E"));</sc
ript> |

```

```

        <a href="../index.php">Домашняя
страница</a>
    </div>
</div>
</body>

```

index3.php

```

<!DOCTYPE html>
<html>
<head>
<title>Поиск библиографических ссылок</title>
<meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1">
<meta http-equiv="content-type"
content="text/html; charset=utf-8">

```

```

<link rel="stylesheet"
href="css/scholarPage.css">
<script type="text/javascript"
src="js/ajax.js"></script>
<script
src="https://cdn.staticfile.org/axios/0.18.0/a
xios.min.js"></script>

```

```

                                keyword:
<script>                                encodeURIComponent(kw) //
function string2ascii(str){                                })
    num=[]                                .then(function (response) {
    for(var i=0;i<str.length;i++){                                response.data=eval('(' +response.data+')'
        num.push(str.charCodeAt(i))                                ); //json
    }                                //console.log(response.data);//debug
    return(num.join('_') )
}                                $('search').innerHTML=response.data.html

window.onload=function(){                                ;
    //                                ;
    var f=$('mySearchForm');
    $('keyword').focus();//                                $('url').innerHTML=response.data.url;
                                })
                                .catch(function (error) {
                                console.log(error);
                                });
                                }

    var showSearch=function(){                                //Событие привязки кнопки поиска
        var kw=f.keyword.value;                                $("submit").onclick=showSearch
        var submit=f.submit.value;
        var page=f.page.value;

        //Если ключевые слова не введены,
        не запрашивать ;
        if(''==kw.replace(/\s/g, '')){                                /**/
            alert('Пожалуйста, введите                                //
            хотя бы одно ключевое слово.');
```

```

        f.page.value=page;
        //
        if(f.keyword.value=='') return;
        showSearch();
    }
}
</script>
</head>

<?php
require 'common/function.php';
myLog("Посетите страницу поиска Google");// ?>

<body>
<div id="url_info" class='card announce'
style="display:none;">
    <h1>Диплом</h1><h3>1</h3>
    <b>Работает над красной
линией!</b><br><a href="help.php"
target="_blank" title="Помощь">Справочный
документ</a>.
</div>

<div id='erWeiMa'>
    <img src='images/fxxsEWM.jpg' /><br>
    <b>Публичный аккаунт</b>
</div>

<div class=myWrap>
    <form method="POST" target=""
id="mySearchForm">
        <b>Ключевие слова:</b>
        <input type="text" name="keyword"
id="keyword" placeholder='Введите здесь
ключевое слово'>
        <input type="button"
name="submit2" id='submit' class="btn"
value="Google">

```

```

<center>
    <span class="spacer"></span>
    Страница
    <input type="button" name="previous"
class='btn light' value="&lt;">
    <input type='text' name='page' id='page'
class='page' value="1">
    <input type="button" name="next"
class='btn light' value="&gt;">
</center>

    <div id='url'></div>
</form>
<br>
</div>

<div id='search'>
    <div
style="width:800px;margin:0 auto;">Данный сайт
позволяет осуществить анализ динамики развития
исследований в области компьютерных наук на
основе библиографических ссылок.</div> </div>

<style>
/*
http://a.biomooc.com/search?q=apples&ei=jxWOXe
aOD5DZwALci67gDg&start=10&sa=N&ved=0ahUKEwimku
TDlPHkAhWQLFAKHdyFC-wQ8tMDCIkC
http://a.biomooc.com/search?q=apples&ei=jxWOXe
aOD5DZwALci67gDg&start=20&sa=N&ved=0ahUKEwimku
TDlPHkAhWQLFAKHdyFC-wQ8tMDCIsC
#search{width:1000px; margin:0 auto;}
*/

#url{color:#eee;}
#gs_hdr,#gs_ab_rt,#gs_gb,#gs_lnv,#gs_n,#gs_ftr
,
#gs_hp_main #gs_hp_tsi {display:none;}

#gs_ccl { margin-left: 10px;}

```

```
.logo{display: none;}
</style>
```

```
<?php
include('common/footer.php');
?>
```

index4.php

```
<!DOCTYPE html>
<html>
<head>
<title>Поиск библиографических ссылок</title>
<meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1">
<meta http-equiv="content-type"
content="text/html; charset=utf-8">

<link rel="stylesheet"
href="css/scholarPage.css">
<script type="text/javascript"
src="js/ajax.js"></script>

<script>
window.onload=function(){
    //Получить форму
    var f=$('#mySearchForm');
    $('#keyword').focus();//Получите фокус
автоматически

    //Событие привязки кнопки поиска
    $("#submit").onclick=showScholar;

    //Выпадающее окно события
    f.since.onchange=f.sortBy.onchange=funct
ion(){
        //Выполните поиск
        if(f.keyword.value=='') return;
        showScholar();
    };
};
```

```
//Предыдущая страница
f.previous.onclick=function(){
    changePage(-1);
}
//Следующая страница
f.next.onclick=function(){
    changePage(1);
}
//Функция перелистывания страниц
function changePage(num){
    var page=f.page.value;

    num>0?++page:--page;

    if(page<=1) page=1;
    if(page>=10) page=10;

    f.page.value=page;
    //Выполните поиск
    if(f.keyword.value=='') return;
    showScholar();
}
</script>
</head>

<?php
require 'common/function.php';
myLog('Visit Home Page');//Журнал входа ?>

<body>
<div id="url_info" class='card announce'
style="display:none;">
    <h1>Диплом</h1><h3>1</h3>
    <b>Работает над красной
линией!</b><br><a href="help.php"
target="_blank" title="Помощь">Справочный
документ</a>.
</div>
```

```
<div id='erWeiMa'>
    <img src='images/fxxsEWM.jpg' /><br>
    <b>Публичный аккаунт WeChat</b>
</div>
```

```
<div class=myWrap>
<form          method="POST"          target=""
id="mySearchForm">
    <b>Ключевые слова:</b>
    <input      type="text"      name="keyword"
id="keyword"   placeholder='Введите здесь
ключевое слово'>
    <input      type="button"     name="submit2"
id='submit'   class="btn" value="Начать поиск">
<br>
```

```
<center>
Год
<select name="since">
<?php
echo "<option value='0' selected='selected'>--
</option>";
//Получить текущий год
date_default_timezone_set("PRC");
$thisYear = date("Y", time());
//Распечатать список по годам
for($y=$thisYear; $y>1990; $y-- ){
    $opt="<option value='".$y."'>" . $y .
"</option>";
    echo $opt.'\n';
}
?>
</select>
```

```
<span class="spacer"></span>
```

```
Сортировать по
<select name="sortBy">
```

```
        <option                                value="0"
selected="selected">актуальности</option>
        <option value="1">дате</option>
</select>
```

```
<span class="spacer"></span>
Страница
<input      type="button"      name="previous"
class='btn light' value="&lt; ">
<input      type='text'        name='page'      id='page'
class='page' value="1">
<input      type="button"     name="next"       class='btn
light' value="&gt; ">
</center>
</form>
```

```
<span id='schloar'>Поиск ...</span>
```

```
</div>
<style>
#gs_hdr,#gs_ab_rt,#gs_gb,#gs_lnv,#gs_n,#gs_ftr
,
#gs_hp_main #gs_hp_tsi {display:none;}

#gs_ccl { margin-left: 10px;}
</style>
```

```
<?php
include('common/footer.php');
?>
```

open_url.php

```
<?php
header("Content-type: text/html; charset=utf-
8");
error_reporting(E_ALL);
ini_set("display_errors","On");
```

```
require 'common/function.php';
//v1.0
?>

<head>
<title>Версия страницы - v1.0</title>
<link                rel="stylesheet"
href="css/scholarPage.css">
<style>
#myWrap{width:1000px; margin:35px auto; }
#mySearchLogo{
    padding:5px;
    margin:5px auto;
    width:100%;
    font-family:"Microsoft YaHei";
}
#mySearchResult  input[type="text"]{width:60%;
border:1px solid #0096ff;
    height:40px;
    padding:0 7px;
}
#mySearchResult
input[type="submit"]{width:100px;   border:1px
solid #0096ff;
    height:40px;
}

#gs_nml{
    display:none;
}
</style>
</head>

<body>
<div id=myWrap>
<form                method="POST"                target=""
id=mySearchResult                onsubmit="return
checkForm()">
    <b id="mySearchLogo">
    Web URL:
```

```
</b>
    <input        type="text"        name="keyword"
value="<?php
        if(isset($_POST['keyword2']))){
            echo
ascii2str($_POST['keyword2']);
        }
    ?>">
    <input        type="submit"        name="submit"
class=btn value="Load Page">
</form>

<script>
//string2ascii
function string2ascii(str){
    num=[]
    for(var i=0;i<str.length;i++){
        num.push(str.charCodeAt(i))
    }
    return(num.join('_') )
}

//js
function checkForm(){
    var                form                =
document.getElementById('mySearchResult');
    //form['keyword2'].value                =
string2ascii(form['keyword'].value)

    //
    var                temp                =
document.createElement('form');
    temp.action = '';
    temp.method = 'post';
    temp.style.display = 'none';
    //
    opt = document.createElement('input');
    opt.name = "keyword2";
    opt.value                =
string2ascii(form['keyword'].value);
    //
```

```

temp.appendChild(opt);
document.body.appendChild(temp);
temp.submit();//
//
return false;
}
</script>

<?php
/*$url="http://www.google.com.hk/search?ie=UTF-8&q=武汉";
if(isset($_POST['keyword'])){
    $keyword=$_POST['keyword'];
    $btn=$_POST['submit'];
    if('Scholar'==$btn){

        //$url="http://scholar.google.com/search?ie=UTF-8&q=$keyword";

        //$url="http://scholar.google.com.cn/scholar?q=%E7%BB%9F%E8%AE%A1";

        $url="http://scholar.google.com.cn/scholar?ie=UTF-8&q=$keyword";
    }else{

        //$url="http://www.baidu.com/s?ie=UTF-8&wd=$keyword";

        $url="http://www.google.com/search?ie=UTF-8&q=$keyword";
    }
}else{
    $url="http://scholar.google.com.cn/";
};
*/
if(isset($_POST['keyword2'])){
    $url2=$_POST['keyword2'];
    if(trim($url2==""){
        die("Pls input the URL");
    }
}

```

```

    }
}else{
    die("Pls input the URL");
}
//
$url=ascii2str($url2);

    $handle = fopen($url, "rb", false);

    $contents = '';
    if(!$handle){
        die("Ошибка открытия файла!");
    }
    while (!feof($handle)) {
        $contents .= fread($handle, 8192);
    }
    echo $contents;
    fclose($handle);
?>

</div>
</body>

```

open_url2.php

```

<?php
header("Content-type: text/html; charset=utf-8");
error_reporting(E_ALL);
ini_set("display_errors", "On");

require 'common/function.php';
//v1.0
?>

<head>
<title>Страница загрузки - v1.0</title>
<link rel="stylesheet" href="css/scholarPage.css">
<style>

```

```
#myWrap{width:1000px; margin:35px auto; }
#mySearchLogo{
    padding:5px;
    margin:5px auto;
    width:100%;
    font-family:"Microsoft YaHei";
}
#mySearchResult input[type="text"]{width:60%;
border:1px solid #0096ff;
    height:40px;
    padding:0 7px;
}
#mySearchResult
input[type="submit"]{width:100px; border:1px
solid #0096ff;
    height:40px;
}

#gs_nml{
    display:none;
}
</style>
</head>

<body>
<div id=myWrap>
<form method="POST" target=""
id=mySearchResult onsubmit="return
checkForm()">
    <b id="mySearchLogo">
Web URL:
    </b>
    <input type="text" name="keyword"
value="<?php
    if(isset($_POST['keyword2']))){
        echo
ascii2str($_POST['keyword2']);
    }
?>">
    <input type="submit" name="submit"
class=btn value="Load Page">
```

```
</form>

<span class=lighter>Example URL:
https://www.google.com/search?q=Snap+shares+so
ar+back+above+IPO+price+as+turnround+takes+hol
d</span><br>
<script>
//string2ascii
function string2ascii(str){
    num=[]
    for(var i=0;i<str.length;i++){
        num.push(str.charCodeAt(i))
    }
    return(num.join('_') )
}

//js
function checkForm(){
    var form =
document.getElementById('mySearchResult');
    //form['keyword2'].value
string2ascii(form['keyword'].value)

    //Новая форма
    var temp =
document.createElement('form');
    temp.action = '';
    temp.method = 'post';
    temp.style.display = 'none';
    // Новый домен
    opt = document.createElement('input');
    opt.name = "keyword2";
    opt.value
string2ascii(form['keyword'].value);
    //Добавить в структуру документа
temp.appendChild(opt);
document.body.appendChild(temp);
temp.submit();//Отправить новую
временную форму
    //Оригинальная форма не отправлена
return false;
```



```
}
</script>
```

```
<?php
/*$url="http://www.google.com.hk/search?ie=UTF-8&q=武汉";
if(isset($_POST['keyword'])){
    $keyword=$_POST['keyword'];
    $btn=$_POST['submit'];
    if('Scholar'==$btn){

        //$url="http://scholar.google.com/search?ie=UTF-8&q=$keyword";

        //$url="http://scholar.google.com.cn/scholar?q=%E7%BB%9F%E8%AE%A1";

        $url="http://scholar.google.com.cn/scholar?ie=UTF-8&q=$keyword";
    }else{

        //$url="http://www.baidu.com/s?ie=UTF-8&wd=$keyword";

        $url="http://www.google.com/search?ie=UTF-8&q=$keyword";
    }
}else{
    $url="http://scholar.google.com.cn/";
};
*/
if(isset($_POST['keyword2'])){
    $url2=$_POST['keyword2'];
    if(trim($url2==""){
        die("Pls input the URL");
    }
}else{
    die("Pls input the URL");
}
//
```

```
$url=ascii2str($url2);
echo $url."<hr>"; //debug
```

```
$ch = curl_init();

//URL
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false); // Проверить источник сертификата
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false); // Проверьте, существует ли алгоритм шифрования SSL из сертификата

//$headers = ['User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36']; //设置一个你的浏览器agent的header
//curl_setopt($ch, CURLOPT_USERAGENT, $_SERVER['HTTP_USER_AGENT']); //Имитация браузера, используемого пользователем
$headers = array(
    #'User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.9) Gecko/2008052906 Firefox/3.0',
    'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36',
    'Referer: https://www.baidu.com',
);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

if (ini_get('open_basedir') == '' && ini_get('safe_mode' == 'Off')) {
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // Использовать автоматический прыжок
}
curl_setopt($ch, CURLOPT_AUTOREFERER, 1); //
```

```
curl_setopt($ch, CURLOPT_TIMEOUT, 30); //
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 40);

curl_setopt($ch, CURLOPT_HEADER, false); //
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
//

curl_setopt($ch, CURLOPTINFO_HEADER_OUT, true);
//PHP 5.1.3
curl_setopt($ch, CURLOPT_REFERER,
'https://www.baidu.com');
//curl_setopt($ch, CURLOPT_SSLVERSION, 2);
```

```
try {
    //HTML
    $output = curl_exec($ch);

    //Получить код ошибки
    $curlErrno = curl_errno($ch);
    if ($curlErrno) {
        throw new
        Exception(curl_error($ch) . '(' . $curlErrno .
        ')') . date('Y-m-d H:i:s',time()) );
    }

    //
    curl_close($ch);

    //Распечатайте полученные данные
    print_r($output);
} catch (\Exception $e) {
    echo $e->getMessage();
}

/*
    $handle = fopen($url, "rb");
```

```
$contents = '';
if(!$handle){
```

```
        die("Ошибка открытия файла!");
    }
    while (!feof($handle)) {
        $contents .= fread($handle, 8192);
    }
    echo $contents;
    fclose($handle);
}
?>
</div>
</body>
```

index.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Поиск библиографических
    ссылок</title>
    <meta http-equiv="X-UA-Compatible"
    content="IE=edge,chrome=1">
    <meta http-equiv="content-type"
    content="text/html; charset=utf-8">

    <link rel="stylesheet"
    href="css/scholarPage.css">
    <script type="text/javascript"
    src="js/ajax.js"></script>
    <script
    src="https://cdn.staticfile.org/axios/0.18.0/a
    xios.min.js"></script>
    <script
    src="https://cdn.jsdelivr.net/npm/chart.js"></
    script>

    <script>

    function string2ascii(str){
        num=[]
```

```

for(var i=0;i<str.length;i++){
    num.push(str.charCodeAt(i))
}
return(num.join('_') )
}

window.onload=function(){
    //Получить форму
    var f=$('#mySearchForm');
    $('#keyword').focus();//

    function sleep(ms) {
        return new Promise(resolve =>
setTimeout(resolve, ms));
    }

    async function demo(kw,
since,sortBy,page) {
        await sleep(2000);

        var data_amount = [];
        // Sleep in loop

        const labels = [
            1991,
            1992,
            1993,
            1994,
            1995,
            1996,
            1997,
            1998,
            1999,
            2000,
            2001,
            2002,
            2003,
            2004,
            2005,
            2006,
            2007,
            2008,
            2009,
            2010,
            2011,
            2012,
            2013,
            2014,
            2015,
            2016,
            2017,
            2018,
            2019,
            2020,
            2021
        ];

        const data = {
            labels: labels,
            datasets: [{
                label: 'данные/год',
                backgroundColor:
                    'rgb(255, 99, 132)',
                borderColor: 'rgb(255,
                    99, 132)',
                data: data_amount,
            }]
        };

        const config = {
            type: 'line',
            data,
            options: {}
        };

        var myChart = new Chart(
            document.getElementById('myChart'),
            config
        );
    }
}

```

```

        for (let i = 1991; i < 2022; i++) {
            axios.post('gs.php', {
                submit: 'axios',
                since:i,
                sortBy:sortBy,
                page:page,
                keyword:
            encodeURIComponent(kw) //
            })
            .then(function
            (response) {
                response.data=eval('(' +response.data+')'); //

                const csLewisQuote
                = response.data.html;

                const regex =
                /Результатов: примерно [\d\s]+/gu;

                let year =
                csLewisQuote.match(regex);

                if (year.length >
                0) {
                    year[0] =
                    year[0].replace(' ', '');

                    let amount =
                    year[0].match(/[ \d\s*]+/g);

                    let amount2 =
                    parseFloat(amount[0].replace(/\.|\s*/g, ''));

                    data_amount.push(amount2);

                    myChart.data.datasets[0].data.push(amount2);

                    myChart.update();
                }
            }).catch(function
            (error) {
                console.log(error);
            });

            await sleep(5000);
        }
    }

    var showSearch=function(){
        var kw=f.keyword.value;
        var submit=f.submit.value;

        var since=f.since.value;
        var sortBy=f.sortBy.value;
        var page=f.page.value;

        //Если ключевые слова не
        введены, не запрашивать ;

        if(''==kw.replace(/\s/g, '')){
            alert('Пожалуйста, введите
            хотя бы одно ключевое слово.');
```

```

        //keyword:    string2ascii(
encodeURIComponent(kw) ) //
        keyword:
encodeURIComponent(kw) //
    })
    .then(function (response)
{
    response.data=eval('(' +response.data+ ')'); //

    console.log(response.data);//debug

        // const regex = /<div
class="gs_ab_mdw">\w+\s*[\d\.\.]+\s*/g;

    $('search').innerHTML=response.data.html;

    $('url').innerHTML=response.data.url;

        })
        .catch(function (error) {
            console.log(error);
        });
    }

    //
    $("submit").onclick=showSearch

    //

f.since.onChange=f.sortBy.onChange=function(){
    //
    if(f.keyword.value=='')
return;

    showSearch();

};
/**/
//
f.previous.onclick=function(){
    changePage(-1);
}
}
f.next.onclick=function(){
    changePage(1);
}
//
function changePage(num){
    var page=f.page.value;

    num>0?++page:--page;

    if(page<=1) page=1;
    if(page>=10) page=10;

    f.page.value=page;
    //
    if(f.keyword.value=='')
return;

    showSearch();
}
}
</script>
</head>

<?php
require 'common/function.php';
myLog('Посетите      страницу      поиска
Google'); //Журнал входа ?>

<body>
<div id="url_info" class='card announce'
style="display:none;">
    <b>Диплом</b><h2>1</h2>
    <br><a href="help.php" target="_blank"
title="Помощь">Справочный документ</a>.
</div>

```

```
<div class=myWrap>
    <form      method="POST"      target=""
id="mySearchForm">
        <b>Ключевые слова:</b>
        <input  type="text"  name="keyword"
id="keyword"  placeholder='Введите  здесь
ключевое слово'>
        <input  type="button"  name="submit2"
id='submit'  class="btn"  value="Начать поиск">

    <center>
        Год
        <select name="since">
            <?php
            echo  "<option  value='0'
selected='selected'>--</option>";
            //

date_default_timezone_set("PRC");
    $thisYear = date("Y", time());
    //
    for($y=$thisYear; $y>1990; $y-
- ){
        $opt="<option
value='".$y."'>" . $y . "</option>";
        echo $opt.'\n';
        }
        ?>
    </select>

    <span class="spacer"></span>

    Сортировать по
    <select name="sortBy">
        <option  value="0"
selected="selected">актуальности</option>
        <option
value="1">дата</option>
    </select>
```

```
<span class="spacer"></span>
Страница
    <input      type="button"
name="previous"      class='btn      light'
value="&lt; ">
    <input  type='text'  name='page'
id='page'  class='page'  value="1">
    <input  type="button"  name="next"
class='btn light'  value="&gt; ">
    </center>

    <div id='url'></div>
</form>
<div style="max-height: 500px; max-width:
1000px">
    <canvas id="myChart"></canvas>
</div>
<br>
</div>

<div      id='search'>      <div
style="width:800px;margin:0 auto;">Данный сайт
позволяет осуществить анализ динамики развития
исследований в области компьютерных наук на
основе библиографических ссылок.</div> </div>

<style>
    /*
    #search{width:1000px; margin:0 auto;}
    */

    #url{color:#eee;}

    #gs_hdr,#gs_ab_rt,#gs_gb,#gs_lnv,#gs_n,#gs_ftr
    ,
    #gs_hp_main #gs_hp_tsi {display:none;}

    #gs_ccl { margin-left: 10px;}
```

30
1116130.01212-01 12 01

```
.logo{display: none;}  
</style>
```

```
<script>
```

```
</script>
```

```
<?php  
include('common/footer.php');  
?>
```

Міністерство освіти і науки України
Український державний університет науки та технологій
Східний науковий центр транспортної академії наук



TEMPUS: CITISET & SEREIN & CRENG

ТЕЗИ

XV Міжнародної науково-практичної конференції
«СУЧАСНІ ІНФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ, В ПРОМИСЛОВОСТІ ТА ОСВІТІ»
до 100-річчя професора Євгена Мироновича Шафіта

ABSTRACTS

of the XV International Conference
«MODERN INFORMATION AND COMMUNICATION TECHNOLOGIES
ON A TRANSPORT, IN INDUSTRY AND EDUCATION»
for the 100th anniversary of Professor Yevgen Mironovich Shafit

ТЕЗИСЫ

XV Международной научно-практической конференции
«СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ И
КОМУНИКАЦИОННЫЕ ТЕХНОЛОГИИ НА ТРАНСПОРТЕ,
В ПРОМЫШЛЕННОСТИ И ОБРАЗОВАНИИ»
на 100-летие профессора Евгения Мироновича Шафита
16.12.2021 – 17.12.2021

Дніпро
2021

РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ АНАЛІЗУ ДИНАМІКИ РОЗВИТКУ ДОСЛІДЖЕНЬ В ГАЛУЗІ КОМП'ЮТЕРНИХ НАУК НА ОСНОВІ ПОШУКУ БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

Саєнко Е. О., Андрющенко В. О., Український державний університет
науки і технологій, Україна

Під час визначення тематики студентських наукових та кваліфікаційних робіт обрання актуального та перспективного напрямку є вкрай важливим і значною мірою визначатиме подальший фокус зусиль студента. Зацікавленість дослідників тим чи іншим напрямком з часом змінюється і може як зростати, так і зменшуватись, що може свідчити в тому числі і про зменшення перспективності цього напрямку.

В даній роботі пропонується розробка програмних засобів, які виконують пошук наукових статей за певною тематикою та аналізують, як з часом змінюється кількість наукових робіт, кількість цитувань, кількість авторів, що публікуються в даному напрямку. Використання сервісу Google Scholar дозволяє використовувати в якості джерел пошуку сайти видавців, наукових конференцій, спеціалізовані соціальні мережі і різні сайти, де викладаються наукові роботи, в тому числі рецензовані. При цьому всі посилання на повнотекстові версії групуються в єдиний кластер. Розроблений в рамках даної роботи сервіс має веб-інтерфейс, у якому користувач вказує ключові слова та параметри для пошуку. Ці дані зберігаються в часовій таблиці, що створюється під кожен пошуковий запит користувача. Вхідними даними для роботи алгоритму пошуку розробленого веб-сервісу є пошукові параметри, що вносяться в таблицю користувачем.

Графічне представлення результатів вибірки у певній науковій сфері досліджень дозволяє об'єктивно оцінювати сучасні наукові набуток за сферою знань.

У ході експериментального дослідження було виявлено такі обмеження використання сервісу, які втім не впливають суттєво на результат: 1) деякі PDF-файли неможливо конвертувати у текстовий формат. Відповідно з них неможливо правильно отримати метадані, щоб перевірити, чи знайдено потрібну статтю; 2) зустрічаються статті тих самих авторів з дуже схожими назвами, внаслідок чого на етапі перевірки, чи не знайдена потрібна стаття, одна з таких статей може бути помилково прийнята за іншу; 3) деякі статті зустрічаються виключно у збірниках, що представляються одним величезним файлом, що не дозволяє їх аналізувати окремо; 4) помилки у назві статті або її авторах також можуть призвести до помилок пошуку.

На основі вищевикладеного варто зробити наступний висновок: розроблений програмний продукт дозволяє здійснювати аналіз динаміки розвитку досліджень в окресленій галузі. У загальному випадку результати пошуку та аналізу насамперед залежать від поставленого завдання та коректності запиту.