

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

СЕРГІЙЧУК Вікторія Вікторівна

**Програмний додаток пошуку інформації в мережі інтернет
на основі онлайн сервісів пошуку інформації /
Software application for searching information on the Internet
based on online information search services**

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи KI-41
Сергійчук Вікторія Вікторівна

Науковий керівник
К.т.н., Батько Ю.М.

Кваліфікаційну роботу
Допущено до захисту
«___» _____ 20 __ р.

Завідувач кафедри
_____ О.М. Березький

ТЕРНОПІЛЬ - 2021

РЕЗЮМЕ

Кваліфікаційна робота на тему «Програмний додаток пошуку інформації в мережі інтернет на основі Bing Web Search API» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 78 сторінок пояснюючої записки, 14 рисунків, 10 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка програмного додатку для пошуку інформації в глобальній мережі інтернет на основі запитів користувачів.

Методи досліджень базуються на теорії алгоритмів (для аналізу розроблених методів та алгоритмів), алгоритмах семантичного аналізу (для обробки запитів користувачів), технологій об'єктно-орієнтованого програмування (для програмної реалізації спроектованої структури програмного додатку).

В кваліфікаційній роботі на основі аналізу існуючих алгоритмів обробки інформації та алгоритмів пошуку розроблений програмний додаток для аналізу та пошуку інформації в глобальній мережі на основі запитів користувача.

Проведено тестування розробленого програмного додатку на мові програмування Java, що підтвердило доцільність використання існуючих цифрових бібліотек для семантичного аналізу інформації та розробки «клієнт-серверних додатків».

Розроблений програмний продукт є ефективним засобом з простим інтерфейсом, що дозволяє вирішувати проблему отримання, аналізу та пошуку інформації в глобальній мережі інтернет на основі запитів користувачів.

Ключові слова: ІНТЕРНЕТ, СИСТЕМА ПОШУКУ, КОРИСТУВАЦЬКИЙ ЗАПИТ, СЕМАНТИЧНИЙ АНАЛІЗ.

RESUME

Qualification thesis “Software application for searching information on the Internet based on Bing Web Search API” in the specialty 123 "Computer Engineering" "bachelor" education degree contains 78 pages of explanatory notes, 14 figures, 10 tables, 3 appendixes. The volume of graphic material is 2 sheets of A3 format.

The purpose of the qualification work is to develop a software application for searching information on the global Internet based on user requests.

Research methods are based on the theory of algorithms (for analysis of developed methods and algorithms), algorithms of semantic analysis (for processing user requests), technologies of object-oriented programming (for software implementation of the designed structure of the software application).

In the qualification work based on the analysis of existing algorithms for information processing and search algorithms developed a software application for analysis and retrieval of information in the global network based on user requests.

The developed software application was tested in the Java programming language, which confirmed the feasibility of using existing digital libraries for semantic analysis of information and development of "client-server applications".

The developed software product is an effective tool with a simple interface that allows you to solve the problem of obtaining, analyzing and retrieving information on the World Wide Web based on user requests.

Keywords: INTERNET, SEARCH SYSTEM, USER REQUEST, SEMANTIC ANALYSIS.

ЗМІСТ

Перелік умовних скорочень.....	9
Вступ.....	10
1 Принципи та програмні засоби для пошуку інформації в базах на основі онлайн сервісів.....	12
1.1 Основні поняття теорії інформації.....	12
1.2 Основні визначення та структура web технології.....	15
1.3 Програмні засоби пошуку інформації в мережі інтернет.....	22
1.4 Постановка задач кваліфікаційної роботи та шляхи її вирішення.....	26
2. Проектування програмного додатку онлайн пошуку інформації.....	27
2.1 Алгоритми пошуку інформації на основі запитів користувачів.....	27
2.2 Цифрова бібліотека Bing Web Search API.....	33
2.3 Алгоритм роботи програмного додатку онлайн пошуку інформації...	36
3. Програмна система онлайн пошуку інформації на основі текстових запитів..	39
3.1 Структура програмної системи пошуку інформації.....	39
3.2 Підсистеми онлайн пошуку інформації.....	49
3.3 Тестування модуля пошуку інформації на основі текстових запитів..	51
4 Техніко-економічний розділ.....	53
4.1 Розрахунок витрат на розробку програмного додатку.....	53
4.2 Визначення експлуатаційних витрат.....	58
4.3 Визначення економічної ефективності та терміну окупності.....	62
Висновки.....	64
Список використаних джерел.....	65
Додаток А Вихідний текст головної функції програмного модуля.....	69
Додаток Б Довідка про впровадження.....	75
Додаток В Світлокопія виданої публікації.....	76

					КР.КІ.110325/17.00.00.000 ПЗ				
Змн.	Лист	№ докум.	Підпис	Дата					
Розробив	Сергійчук В.В.				ПРОГРАМНИЙ ДОДАТОК ПОШУКУ ІНФОРМАЦІЇ В МЕРЕЖІ ІНТЕРНЕТ НА ОСНОВІ BING WEB SEARCH API	Літ.	Арк.	Акрушів	
Перевір.	Батько Ю.М.						8	78	
Консульт.						ЗУНУ, ФКІТ, КІ-41			
Н. Контр.	Мельник Г.М.								
Затвердив	Березький О.М.								

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

КС	–	Комп’ютерна система
БД		База даних
UGI	–	User Graphics Interface
CAT	–	Семантичний аналізатор тексту
ПЗ	–	Програмне забезпечення

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Більшість інформації можна знайти в Інтернеті за допомогою пошукових систем. Пошукова машина - це веб-служба, яка використовує веб-роботи для запиту мільйонів сторінок в Інтернеті та створює індекс цих веб-сторінок. Потім користувачі Інтернету можуть використовувати ці послуги для пошуку інформації в Інтернеті. Пошукові системи особливо корисні, коли ви не знаєте URL у вигляді веб-сайту, але потрібно знайти його. Шукаючи інформацію в Інтернеті, пам'ятайте наведені нижче поради, щоб покращити пошук.

Якщо ви шукаєте конкретну фразу, наприклад, комп'ютерну довідку, розмістіть навколо фрази лапки, щоб отримати результати для саме цього поєднання слів. Наприклад, введіть "комп'ютерна довідка" як критерій пошуку для відповідності сторінок, де ці два слова з'являються разом.

Цей фокус також можна використовувати в частинах вашого пошукового запиту. Наприклад, "комп'ютерна довідка" Майкрософт шукатиме все, що містить Microsoft і комп'ютерну допомогу, разом.

Ви можете включити кілька пошукових фраз у свій пошук. Наприклад, пошук за "комп'ютерною довідкою" "Microsoft Windows" дасть результати для сторінок, які містять обидві ці точні фрази.

Багато пошукових систем вилучають загальні слова, які вони називають словами-зупинками для кожного виконаного пошуку. Наприклад, замість того, щоб шукати, чому мій комп'ютер не завантажується, пошукова машина шукатиме комп'ютер і завантажуватиметься. Щоб запобігти видаленню цих слів, оточіть пошук цитатами.

Багато пошукових систем дозволяють операторам логічної логіки допомагати відфільтрувати погані результати. Хоча загальноприйняті логічні значення включають "і", "або", і "ні", більшість пошукових систем замінили ці ключові слова символами. Наприклад, щоб знайти комп'ютерну довідку без результатів, що містять Linux, слід набрати комп'ютерну довідку -linux. "-Linux"

вказує пошуковій системі виключити будь-які результати, що містять слово Linux.

Багато пошукових систем дозволяють використовувати додатковий синтаксис, який допомагає обмежити ваші пошукові рядки. Наприклад, Google дозволяє користувачам шукати посилання на певну сторінку, набираючи "link:" та інші ключові слова на початку пошукового запиту.

Проте це все просто тільки на перший погляд, оскільки в середині вже готових сервісів вмонтовано десятки алгоритмів, які проводять аналіз, поділ та розбиття вхідного запиту на множину ключових слів, а в подальшому дозволяють у більшій мірі та з меншою надлишковістю отримувати відповіді на користувацькі запити.

Пошукові системи ставлять найрелевантніші результати на перше місце, так що якщо ви хочете немає в списку найкращих, спробуйте ще раз із більш конкретними ключовими словами. Або звужте пошук, додавши більше ключових слів. Google повертається лише сторінки, що містять усі ключові слова, які ви використовуєте, тому додайте більше ключові слова усувають менш релевантні сторінки. Щоб отримати більше результатів, використовуйте більш загальні ключові слова або включають менше пошукових термінів. Посилання Google "в кеш" показує вам знімок веб-сторінки може бачити його, навіть якщо фактичний сайт недоступний. На кешованому сторінки, ваші ключові слова виділяються, щоб їх було легше знайти. Перегляд місця, де ваші ключові слова відображаються на веб-сторінці до натискання посилання на результат може заощадити ваш час. Ось чому Google відображає витяг із кожної повернутої сторінки результатів, що відображає ваш запит терміни жирним шрифтом. Ці фрагменти дозволяють побачити контекст, в якому ваші пошукові терміни відображаються на сторінці, тому ви можете визначити, чи є сайт варто відвідати.

Створення алгоритму є дуже важливим етапом програмування, оскільки воно визначає ефективність програми. Тому задача створення програмного додатку аналізу запитів та пошуку інформації є актуальною.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1 ПРИНЦИПИ ТА ПРОГРАМНІ ЗАСОБИ ДЛЯ ПОШУКУ ІНФОРМАЦІЇ В БАЗАХ НА ОСНОВІ ОНЛАЙН СЕРВІСІВ

1.1 Основні поняття теорії інформації

Класифікація інформації, як випливає з назви, є процесом класифікації інформації за відповідними категоріями. Ми обробляємо незліченну кількість інформації щодня, навіть не усвідомлюючи цього. Інформація - це те, що диктує прийняття рішень, починаючи від найпоширеніших рішень (яку книгу я повинен прочитати) до дуже критичних, наприклад (куди вкласти гроші). Наявність такого роду конфіденційної та важливої інформації може надати організації повноваження та одночасно відповідальність за захист від вразливостей до загроз безпеці. Ось чому захист інформації є дуже важливим процесом, яким організації намагаються успішно керувати. І першим кроком успішного захисту інформації є класифікація інформації.

Класифікація інформації може здатися легкою, але коли говоримо про інформацію великого обсягу, різноманітності та важливості, виконання цього завдання стає набагато складнішим. Є три кроки, які полегшують виконання цього процесу:

- знати свої інформаційні активи та присвоювати значення кожному з них;
- позначте кожен інформаційний актив;
- метод обробки кожного інформаційного активу.

Для ефективної класифікації інформації організація повинна присвоїти значення кожному інформаційному активу відповідно до ризику втрати або шкоди від розголошення. Відповідно інформацію слід сортувати за категоріями, поясненими нижче:

Конфіденційна інформація - це будь-яка інформація, яка зберігається як конфіденційна усіма сторонами, включеними в цю інформацію або на них впливає. Іноді терміни конфіденційна інформація та секретна інформація використовуються в одному контексті; однак секретна інформація насправді частіше використовується державними установами як юридичний термін.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Класифікована інформація - це конфіденційна інформація, доступ до якої обмежений законом або нормативними актами. Якщо будь-яка сторона володіє секретною інформацією, для обробки такої інформації вимагається офіційне підтвердження безпеки.

Інформація з обмеженим доступом - представляє всю інформацію, доступну більшості працівників, але не всім.

Внутрішня інформація - це інформація, до якої мають доступ усі працівники.

Публічна інформація - це інформація, до якої мають доступ усі в організації та за її межами.

Як тільки організація класифікує свою інформацію на основі її вартості, власник активу повинен створити систему або формат для маркування даних. Незалежно від того, зберігаються дані фізично чи в цифровому вигляді, маркування повинно бути простим, легким для розуміння, а головне, послідовним. Цифрові файли можна позначати як у цифровому, так і в алфавітному порядку, якщо вони систематичні, надійні та прості в дотриманні. Додавання візуальних ярликів до верхнього та нижнього колонтитулів файлів може підвищити обізнаність та допомогти співробітникам стати уважнішими до безпеки та уникнути обміну вмістом на USB-накопичувачах, електронній пошті або за допомогою хмарних служб.

Нарешті, після того, як організація класифікує та маркує всі свої інформаційні активи, вона повинна створити набір правил та намітити спосіб захисту своєї інформації на основі класифікації. Наприклад, публічна інформація може бути розміщена у відкритому кабінеті або опублікована на соціальних медіа-платформах компанії, в той час як секретна інформація повинна зберігатися в заблокованому та захищеному вигляді, або на безпечному сервері, або фізично відстежуватися фахівцями з безпеки.

Існує чотири основні причини, через які класифікація інформації важлива:

- ефективність;
- безпека;
- культура безпеки;

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

— відповідність.

Ефективність. Організації, які мають свою класифіковану інформацію, можуть ефективніше доставляти та виконувати щоденні операції. На основі їх класифікації дані можна легко знайти, а зміни легко простежити.

Безпека. Безпека - це основна ідея класифікації інформації. Наявність знань про те, які дані ви зберігаєте, полегшує надійний захист конфіденційних даних. Усі організації в наш час керуються даними, і їх дані цінні, знають вони це чи ні, оскільки це дуже важлива частина їх діяльності, а часто і самого існування. Отже, шифрування даних, зберігання їх на безпечних серверах із надійними брандмауерами, що відповідає різним стандартам захисту даних, може бути чудовою підмогою для запобігання зовнішнім загрозам.

Окрім загроз, які можуть надходити ззовні, іноді всередині нас є ті, про які слід турбуватися. Серед найскладніших запобігань загрози інсайдерів походять від слабких місць працівників. Вони можуть бути пустотливими, включаючи навмисне викрадення даних або навіть випадкові порушення даних. Ось чому обмеження інформації та класифікація інформації відіграють велику роль у питанні запобігання внутрішнім загрозам. У поєднанні з тренінгами та технологією управління доступом класифікація інформації допомагає запобігти порушенням даних.

Культура безпеки. Впровадження Класифікації інформації допомагає формувати культуру обізнаності щодо безпеки в організації. Він покладає відповідальність за захист інформації на кожного, хто обробляє її, і гарантує, що всі працівники розуміють цінність інформації, з якою вони працюють щодня, і знають, як з нею поводитися.

Працівники повинні отримувати доступ до документів на основі необхідності знання. Ця система може намітити привілеї доступу працівників на основі рівня чутливості даних документа, полегшуючи відстеження та запобігаючи будь-якому неправильному використанню або маніпулюванню інформацією.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Основні визначення та структура web технології

На відміну від традиційних джерел інформації, таких як телебачення, газети та журнали, Інтернет забезпечує споживачів більш інтерактивним спілкуванням, що є його унікальною характеристикою як засобу масової інформації. Таким чином, аналіз того, як споживачі використовують Інтернет як нове джерело інформації, стає цікавою темою як для дослідників, так і для практиків. Зростаюча залежність Інтернету від пошуку інформації зумовлена наступними перевагами:

- низькі трансакційні витрати;
- простіший доступ до інформації про ціни та товари;
- зручне придбання супутніх послуг та можливість об'єднання послуг.

Порівняно з епохою без Інтернету, споживачі можуть точніше приймати рішення про покупку через велику кількість джерел інформації в Інтернеті. При класифікації товарів на продукти пошуку та досвіду значення різних джерел інформації про товари для споживачів різняться. Відповідно пошукові товари визначаються як ті, де переважають атрибути товару, щодо яких можна отримати повну інформацію перед покупкою, наприклад спортивне взуття чи мобільні телефони. У описі продуктів переважають атрибути, які неможливо дізнатись до моменту придбання та використання продукту, або для яких пошук інформації є більш дорогим та/або складним, ніж прямий досвід роботи з продуктами, наприклад, туристичні пакети та вечері в нових ресторанах. Крім того, інформаційні витрати споживачів на пошук значно відрізняються залежно від характеристик продуктів пошуку та досвіду. Внутрішні атрибути пошукових продуктів є легкодоступними, конкретними та більш об'єктивними для порівняння якості товару. На відміну від цього, споживачі покладаються на більш незвичні підказки, щоб оцінити якість досвіду продуктів або послуг. Наприклад, інформація для вибору туристичного пакету є більш абстрактною та орієнтованою на досвід. Отже, рекомендації інших будуть використовуватися більше для досвіду, ніж для продуктів пошуку.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Що стосується Інтернету, цей інтерактивний засіб відкриває можливості онлайн-пошуку інформації як альтернативи досвіду. Тепер споживачі можуть легко отримати інформацію з досвіду інших людей щодо цих продуктів. Ставлення споживачів та використання джерел інформації в Інтернеті, включаючи рекламу веб-сайтів, веб-сайти бізнесу та роздрібної торгівлі, онлайн-ові групи новин, спільноти та чати. Основні аргументи цього дослідження полягають у тому, що споживачі досвідчених продуктів повинні цінувати та використовувати інформацію в Інтернеті більше, ніж інформацію пошукової продукції, через характеристики різних продуктів та співвідношення витрат/вигод різних джерел інформації.

На сьогоднішній день існує безліч веб-технологій, проте для аналізу їх було погруповано в класи які найчастіше використовуються як початківцями, так і експертами в цій галузі. Було виділено наступні класи (рисунок 1.1):

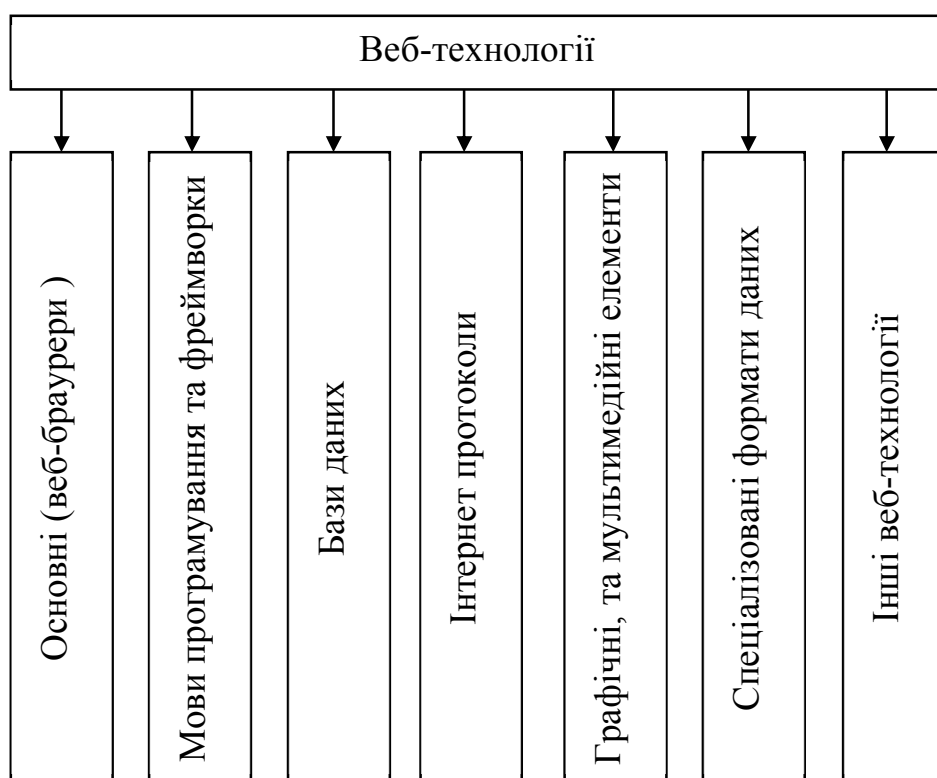


Рисунок 1.1 – Класифікація основних веб-технологій

Веб-технології - це різні інструменти та техніки, які використовуються в процесі спілкування між різними типами пристроїв через Інтернет. У цьому випадку Інтернет стосується Всесвітньої павутини, більш відомої як WWW.

Вперше він з'явився у 1989 році, коли відомий вчений та інженер Тім Бернерс-Лі створив ефективний механізм розподілу ресурсів між науковцями усього світу.

Основи, які охоплюватимуть веб-браузери та деякі основи розробки веб-додатків. Мови програмування та фреймворки, які використовуються при розробці веб-сайтів. Бази даних, які використовуються на сервері для зберігання даних, необхідних або зібраних веб-сайтами. Деякі протоколи, тобто правила спілкування в Інтернеті. Графічні, аудіовізуальні та інші мультимедійні елементи. Деякі формати даних, які зазвичай використовуються для передачі даних через Інтернет. Інші різні веб-технології.

Веб-браузери, які просто називають браузерами, дають нам змогу переглядати всі ресурси, що є частиною Всесвітньої мережі. Вони засновані на архітектурі клієнт-сервер. У цьому випадку клієнтом є браузер. Ви можете сприймати сервер як комбінацію програмного та апаратного забезпечення, яке отримує запити клієнта, а потім надсилає запитуваний ресурс клієнту.

Кожного разу, коли ви вводите URL-адресу в адресний рядок браузера, він передає ваш запит серверу, а потім отримує та відображає те, що користувач просив.

Веб-розробка відноситься до процесу створення веб-сайтів. Цей процес базується на ряді етапів, які ми детальніше розглянемо, коли вивчимо інструменти, що беруть участь у кожному процесі.

Frontend стосується всіх тих частин веб-сайту, які користувач може бачити на своєму екрані та взаємодіяти.

Бекенд стосується прямо протилежного цьому. Він включає приховані механізми, які роблять функцію веб-сторінки. Типовий користувач, як правило, не знає, що відбувається у серверній системі.

Дані технології хоча і використовуються в загальному для створення програмних продуктів в одному напрямку. Проте несуть зовсім різні навантаження, оскільки дозволяють розробникам розділити свої зусилля по створенню кінцевого продукту. Приклад різниці між цими двома технологіями наведено в таблиці 1.1

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Таблиця 1.1 – Порівняння технологій веб-розробок

Frontend	Backend
З боку клієнта	На стороні сервера
Дизайн веб-сайту	Бази даних
UI / UX	Сервери
Деякі технології інтерфейсу користувача:	Деякі внутрішні технології:
HTML	PHP
CSS	Java
JavaScript	Python
AJAX	Рубін
	.NET

Мова розмітки HyperText, яку частіше називають HTML, - це місце, де розпочався веб-сайт. Це найважливіша складова всіх вступних уроків, які охоплюють основи роботи в Інтернеті. Це мова клієнта, яка використовується для кодування інтерфейсу веб-сайту. Це допомагає розробникам визначити структуру веб-сторінки, і робить це за допомогою елементів, які ідентифікуються тегами.

CSS - одна з найбільш фундаментальних технологій дизайну веб-сайтів. Щоб ще більше прикрасити ваші веб-сторінки, ви можете інтегрувати CSS (каскадні таблиці стилів) у свій HTML-код.

Процес застосування певних стилів до певних елементів HTML-коду може перетворитися на повторюване завдання. Для економії часу та енергії CSS можна використовувати для впорядкування процесу стилізації сторінок на всьому сайті. Після того, як ви підготуєте таблицю стилів із усіма типовими стилями шрифтів, кольорами та іншими характеристиками, пов'язаними із загальним макетом веб-сторінки, ви можете застосувати її для всього сайту. CSS також поєднується з Bootstrap для розробки перших мобільних веб-програм.

Мови програмування - це найосновніший компонент створення веб-сайтів. Для того, щоб кодувати веб-сайт, повинні бути знайомі з декількома концепціями програмування, деякими мовами та їх синтаксисом, а також з хорошою розробкою IDE . Деякі з цих мов можуть бути загального призначення, але інші спеціально створені з метою веб-розробки.

Простіше кажучи, фреймворк можна визначити як скелетний код. Подумайте про це як про грубий контур без дрібних деталей. Структура визначає основні функції або завдання, які передбачається виконувати кодом. Однак для того, щоб розповісти цій програмі, як виконувати ці функції, вам доведеться написати кілька більш детальних, конкретних інструкцій.

Вибір серед цих варіантів залежить від ваших потреб у розвитку. Вам потрібно визначити, яка структура та комбінація мов найкраще відповідатимуть вашим вимогам до масштабованості, безпеки та продуктивності, не вкладаючи багато коштів у вивчення самої технології.

Усі дані, якими обмінюються в Інтернеті, потрібно десь зберігати. Для цього на більшості веб-сайтів пов'язані власні бази даних.

Нижче наведено перелік деяких баз даних – деякі реляційні та інші, нереляційні - які зазвичай використовуються для веб-додатків.

- MySQL;
- SQL Server;
- Postgres;
- Oracle;
- MongoDB.

Кожного разу, коли виникає потреба обмінюватися даними між двома пристроями в Інтернеті, дотримується належної процедури. Дані упаковані належним чином для передачі від джерела до пункту призначення. Спеціальні API (інтерфейси програмування програм) розроблені та інтегровані в веб-сайти для зручного обміну даними. Вони розташовують дані таким чином, що приймач може легко їх декодувати та зрозуміти.

Нижче наведено два поширені формати даних, що використовуються при веб-розробці.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

XML розшифровується як Extensible Markup Language. Синтаксис HTML ми обговорювали на початку цієї статті. XML працює подібним чином, оскільки розбиває дані на елементи, ідентифіковані різними типами тегів. Однак за допомогою XML ви можете вигадувати власні теги, щоб краще описувати ваші дані. Ці дані, потрапивши до веб-програми або сервера, можна легко зрозуміти та проаналізувати.

Після багатьох років випробувань гоміздкості XML та великого споживання пропускнуої здатності, експерти придумали JSON (JavaScript Object Notation). На відміну від XML, JSON зосереджується більше на швидкому та простому обміні даними, а не на детальному визначенні та моделюванні даних. Це також усуває зайве навантаження, яке несе XML у вигляді повторюваних тегів.

JSON базується на звичному механізмі пари ключ-значення. Його синтаксис нагадує JavaScript, що робить його досить простим для читання та розуміння.

Веб-протоколи – це деякі заздалегідь визначені правила, яких повинен дотримуватися кожен, хто спілкується через Інтернет.

Протокол передачі гіпертексту, більш відомий як HTTP, - це веб-протокол, який визначає два поняття:

У запиті повинні бути зазначені наступні чотири пункти:

- url-адреса ресурсу, який бажає клієнт;
- метод, такий як GET (який використовується для отримання даних із сервера) або POST (який використовується для надсилання даних, які потрібно оновити на сервері);
- список заголовків (заголовки запитів можуть містити різну інформацію про клієнта або ресурс, який вони запитували, наприклад ім'я браузера клієнта, операційну систему, розмір повідомлення, час і дату запиту, інформацію про контроль доступу/дані тощо);
- тіло, яке містить інформацію, яку клієнт хоче надіслати на сервер.

Після обробки запиту сервер надсилає відповідь з іншого кінця. Ця відповідь визначає три елементи:

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

- код стану у формі тризначного номера (код стану, який ви могли бачити під час щоденних сеансів веб-перегляду, - 404, що вказує на те, що запитаний ресурс не вдалося знайти);
- список заголовків (заголовки відповідей за структурою нагадують заголовки запитів і можуть містити таку інформацію, як розмір повідомлення, тип вмісту тощо);
- тіло, яке містить інформацію, яку клієнт запитував від сервера.

Таблиця 1.2 – Приклад веб-протоколів

Протокол	Відповідає за	Використання
TCP	Протокол управління передачею	Для встановлення зв'язку між двома пристроями в Інтернеті та управління доставкою пакетів даних
IP	Інтернет-протокол	Для передачі пакетів даних між двома пристроями в Інтернеті
TCP / IP		Ці дві веб-технології часто працюють разом.
UDP	Протокол датаграми користувача	Для встановлення зв'язку між двома пристроями в Інтернеті
FTP	Протокол передачі файлів	Для передачі файлів між клієнтом та сервером
SMTP	Простий протокол пересилання пошти	Для відправки повідомлень електронної пошти на сервер
POP	Протокол поштового відділення	Для отримання повідомлень електронної пошти з сервера
SOAP	Простий протокол доступу до об'єктів	Для надсилання XML-повідомлень через HTTP

Графічні елементи - ключова особливість будь-якої веб-сторінки. Вони не тільки сприяють покращенню сторінки, але й можуть бути використані для передачі важливих моментів кращим чином, ніж текст.

1.3 Програмні засоби пошуку інформації в мережі інтернет

Пошукова система є програмне забезпечення з доступом в Інтернет, яка шукає в базі даних інформації в відповідності з користувачем запиту. Механізм надає список результатів, які найкраще відповідають тому, що користувач намагається знайти. Сьогодні в Інтернеті доступно багато різних пошукових систем, кожна з яких має свої можливості та можливості. Першою коли-небудь розробленою пошуковою системою вважається Арчі, яка використовувалася для пошуку файлів FTP, а першою пошуковою системою на основі тексту вважається Вероніка. В даний час найпопулярнішою та найвідомішою пошуковою системою є Google. Інші популярні пошукові системи включають AOL, Ask.com, Baidu, Bing, DuckDuckGo та Yahoo.

Для користувачів доступ до пошукової системи здійснюється через браузер на їх комп'ютері, смартфоні, планшеті чи іншому пристрої. Сьогодні більшість нових браузерів використовують універсальне вікно пошуку - текстове поле у верхній частині браузера. Омнібокс дозволяє користувачам вводити URL-адресу або пошуковий запит.

Оскільки великі пошукові системи містять мільйони, а іноді і мільярди сторінок, багато пошукових систем відображають результати залежно від їх важливості. Це значення зазвичай визначається за допомогою різних алгоритмів.

Як показано, джерело всіх даних пошукової системи збирається за допомогою павука або сканера, який збирає її інформацію (рисунки 1.3).

Після сканування сторінки дані, що містяться на сторінці, обробляються та індексуються. Часто це може включати наступні кроки:

- видаляються стоп-слова;
- зберігають решту слів на сторінці та частоту їх появи;
- зберігають посилання на інші сторінки;
- записується інформація про будь-які зображення, аудіо та вбудовані носії на сторінці.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

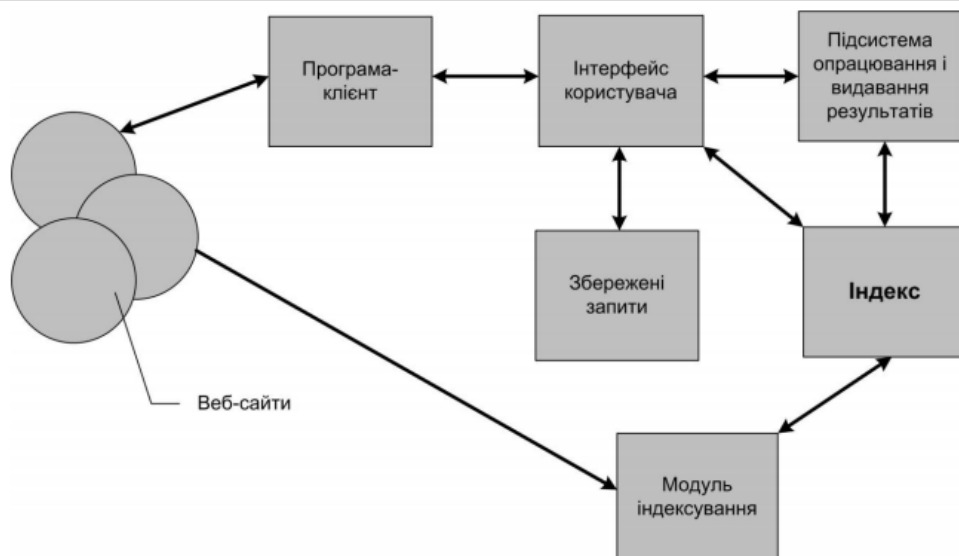


Рисунок 1.2 – Узагальнена структура програмного додатку онлайн пошуку інформації

Стоп-слова – це загальновживані слова, які виключаються з пошуку, щоб швидше індексувати та аналізувати веб-сторінки. Хоча більшість пошукових систем в Інтернеті та NLP (обробка на природній мові) використовують слова зупинки, вони не заважають користувачеві використовувати їх, але вони ігноруються. Деякі загальні слова зупинки включають наступне: а, і, але, як, або, що, буде тощо.

Наприклад, якщо проводиться пошук "Що таке материнська плата?" на якомусь інтернет аукціоні, то пошукова машина шукає лише термін "материнська плата".

Взагалі кажучи, більшість стоп-слів - це функціональні (наповнювальні) слова, які є словами з незначним значенням або зовсім не мають значення, що допомагає сформулювати речення. Слова, що містять зміст, такі як прикметники, іменники та дієслова, часто не розглядаються як слова зупинки. Проте програміст може додати дуже поширені слова. Наприклад, система може вважати "комп'ютер" стоп-словом, оскільки його можна використовувати для опису будь-якого пов'язаного з комп'ютером продукту (наприклад, материнської плати комп'ютера, відеокарти комп'ютера тощо).

Немає універсального списку зупинних слів, які використовує кожна компанія чи програміст. Як і більшість рішень, використання зупинних слів не є ідеальним рішенням і може спричинити будь-яку з наступних ситуацій:

- видалення стоп-слів, які входять до складу товару, компанії чи іншої назви. Наприклад, вилучення слова "дійсно" із "справді простого синдикування" та перетворення недійсного пошуку на справді просте синдикування;

- не видаляють стоп-слова, які є множинними або мають розділові знаки;

- видалення стоп-слова, що є частиною неправильно написаного слова.

Наприклад, пошук "що року", видалення "що" як стоп-слова;

- не видаляти стоп-слово, оскільки воно неправильно написане.

Наприклад, "wat is stop" стає "wat stop", оскільки "what" було написано неправильно.

Зібрані дані використовуються для ранжирування кожної сторінки. Потім ці рейтинги визначають, які сторінки відобразатись у результатах пошуку та в якому порядку.

Нарешті, як тільки дані обробляються, вони розбиваються на файли, вставляються в базу даних або завантажуються в пам'ять, де вони отримують доступ під час пошуку.

Пошукові системи використовують власні алгоритми для індексації та співвіднесення даних, тому кожна пошукова система має свій підхід до пошуку того, що намагаєтеся знайти. Її результати можуть базуватися на тому, де знаходитесь, що ще шукали та яким результатам віддавали перевагу інші користувачі, які шукали те саме. Кожна пошукова система однозначно зважає їх і пропонує різні результати.

Немає жодної пошукової системи, яка б була кращою за всі інші. Багато людей можуть стверджувати, що пошукова система Google - найкраща, найпопулярніша та найвідоміша. Це настільки популярно, що люди часто використовують його як дієслово, коли кажуть комусь шукати їх питання.

Пошукова система Microsoft Bing також популярна і використовується багатьма людьми. Бінг чудово справляється з пошуком інформації та

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

відповідями на запитання. Bing - це те, що забезпечує пошук у Windows 10 та пошуковій системі Yahoo .

Користувачі, які турбуються про конфіденційність, користуються Duck Duck Go. Ця пошукова система робить своїх користувачів анонімними та є чудовим рішенням для користувачів, які цікавляться тим, скільки інформації Google та Bing збирають на своїх користувачів.

Результати порівняння різних програмних засобів наведено в таблиці 1.3:

Таблиця 1.3 – Порівняння програмних онлайн пошуку інформації

Назва програми	Мова інтерфейсу	Розширений пошук	Автоматичний переклад	Додаткові параметрів	Доступ
AOL	англ	+	-	+	free
Ask.com	англ	+	-	-	free
Baidu	англ	+	-	-	free
Bing	англ	+	+/-	+	free
DuckDuckGo	англ	+	-	-+	free
Yahoo	на вибір	+	+/-	+	free
META	укр	+	-	+	free
ukr.net	на вибір	+	-	+	free
i.ua	укр	+	-	-	free
Google	на вибір	+	+	+	free

Багато компромісів щодо інтелектуальної власності можна було б уникнути, якщо б було запроваджено класифікацію інформації. Класифікація інформації допомагає гарантувати, що особи, які беруть участь у організації, мають знання та обізнаність про тип даних, з якими вони працюють, та їх цінність, а також про їхні обов'язки та відповідальність щодо їх захисту та запобігання їх пошкодженню або втраті.

Класифікація інформації - не єдине рішення, яке захищає інформацію або забезпечує відповідність нормативним вимогам. Але це дуже важливий крок на шляху серйозного ставлення до безпеки даних.

1.4 Постановка задач кваліфікаційної роботи та шляхи її вирішення

В розділі було проаналізовано основні поняття інформації та виділено основні характеристики для подальшого її опрацювання. Розглянуто технологію web, основні моделі, визначення та структури, що дозволило оцінити складність запитів, що необхідно буде передавати при роботі з онлайн сервісами. Досліджено онлайн сервіси пошуку інформації, що дало можливість виділити їх переваги та недоліки.

Метою кваліфікаційної роботи є розробка програмного додатку для пошуку інформації в глобальній мережі інтернет на основі запитів користувачів. Розроблений програмний додаток дозволяє опрацьовувати користувацькі запити та проводити на основі них пошук в глобальній мережі інтернет.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- 1) дослідити та провести класифікацію видів інформації;
- 2) проаналізувати можливості web технологій при передачі великих об'ємів даних;
- 3) провести аналіз існуючих програмних додатків для онлайн пошуку інформації;
- 4) провести аналіз сучасних алгоритмів пошуку інформації ;
- 5) спроектувати та провести моделювання структури програмного додатку пошуку інформації;
- 6) програмно реалізувати систему онлайн пошуку інформації на основі користувацьких запитів, провести її тестування та порівняльний аналіз з програмами-аналогами.

2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ ОНЛАЙН ПОШУКУ ІНФОРМАЦІЇ

2.1 Алгоритми пошуку інформації на основі запитів користувачів

Тенденції зростання інформації – це глобальні проблеми, загальні для всіх. У цей інформаційний вік обсяг генерування інформації та її поширення в багатьох формах створили проблеми зберігання, обробки та розповсюдження інформації. Бібліотеки та інші сфери послуг намагалися подолати ці проблеми, застосовуючи кілька сучасних методів. Поступово концепція «цифрової бібліотеки» виникла як провідне технологічне вирішення постійних проблем розширення доступу, тривалого архівування та поширення інформації. Однак цифрова бібліотека із зближенням Інтернету - запропонувала нам потужний засіб управління електронними ресурси. Поняття "Бібліотека без стін" майже стало реальністю завдяки Інтернету, який є єдиним найбільшим джерелом інформації в історії людства, і кидає виклик традиційним уявленням про права на інформацію. Практично існує величезна кількість джерел, які мають відношення до запитів будь-якого користувача, але незліченні стеки веб-ресурсів різко створюють перешкоду для ефективного та ефективного доступу до інформації. Насправді існує значний шум при отриманні інформації - як величезна кількість різнорідних ресурсів, доступних у великому кіберпросторі. Отже, Інтернет спричинив різкі зміни в інформаційному бізнесі та феноменальне збільшення ресурсів в Інтернеті вимагає ефективного методу отримання інформації . Хоча семантична пошукова система та мета-пошуковий механізм стимулюють пошук ресурсів у цифрових колекціях. У цьому контексті різні алгоритми пошуку мають суттєвий вплив на те, як Індекс індексується, і поліпшить пошук ресурсів в Інтернеті. Ці алгоритми відрізняються за можливостями задоволення потреб користувачів, і дуже складно встановити загальний алгоритм для отримання цифрової інформації. Без сумніву, алгоритми пошуку є неминучим компонентом будь-якого механізму отримання інформації, що стосується інформації з цифрової колекції. Більше того, ідентифікація та пошук цифрової інформації

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

визначається ефективністю пошукових служб, але більшості пошукових працівників недостатньо для того, щоб висловити свою інформацію в реальному запиті. Для звичайного користувача Інтернету пошук обмежується цим пошуковим механізмом, використовуючи типовий пошук за ключовими словами, але індекси та павуки / роботи занадто погано перекладають ресурси у репрезентативні ключові слова. Якщо насправді більшість інтернет-серферів не знають абсолютного потенціалу цього. Тому їх пошуки призводять до отримання великої кількості сторінок, де більшість пошуків практично не мають значення, а майже всі відповідні сторінки залишаються прихованими або невидимими для механізму пошуку. Відомо, що близько 85% користувачів ведуть пошук лише на першому екрані результатів пошуку. Вище обговорено питання про появу ефективного механізму пошуку. У цьому плані ручні методи (індексація та пошук людей) є дуже трудомісткими та обмежувачими, коли задіяні великі бази даних або динамічні сторінки. Проблеми традиційних методів призвели до зростання інтересу до методів автоматичного пошуку інформації, що створює виклик традиційному пошуку інформації (IP). Але при автоматизованому пошуку не існує послідовної практики індексування та фільтрації, щоб забезпечити якість та надійність пошуку значно спрощує спілкування між членами команди на етапі проектування чи розв'язку деякої задачі.

„Пошук” - це систематичне вивчення інформації в базі даних з метою ідентифікації об'єктів або об'єктів, які відповідають певним попередньо встановленим критеріям. В іншому випадку пошук означає операцію пошуку конкретного об'єкта в заданій послідовності об'єктів „n”. Тут кількість часу, необхідного для пошуку певного об'єкта чи поля даних чи частини інформації з магазину, називається «часом пошуку». Варто зазначити, що сортування є допоміжним засобом для пошуку. Слід також сказати, що кожен елемент містить у собі частину інформації, яка називається «ключем», а даний ключ пошуку називається «ключем пошуку». Технічно, професіонал у галузі бібліотеки та інформації повинен розрізняти дві різні задачі при обробці інформації - процес зберігання (введення ключа в магазин, використовуючи певний алгоритм зберігання) та процес пошуку (доступ до ключа з магазину). У процесі пошуку

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

можна шукати, що відповідає даному ключу, використовуючи певний алгоритм пошуку. Тепер термін Інформаційне отримання в основному стосується структури, аналізу, організації, зберігання, пошуку та розповсюдження інформації. Вважається що є 6 основних етапів роботи таких систем:

- вибір документів;
- індексація;
- словниковий запас;
- пошук
- інтерфейс користувацької системи
- узгодження.

Знову ж ці підсистеми можуть бути об'єднані у дві широкі категорії, такі як „предмет або вміст” - передбачає аналіз, організацію та зберігання інформації, а „стратегія пошуку” - передбачає керіаналіз користувачів, створення формул пошуку та фактичний пошук. Технічно, алгоритм повинен досягти результату через кінцеву кількість кроків, таким чином виключаючи «методи грубої сили для певних проблем, хоча деякі можуть стверджувати, що пошук грубої сили був також дійсним (загальним) алгоритмом. Цей термін вільно використовується для будь-якої послідовності дій (які можуть припинятися або не закінчуватися). Отже, алгоритм - це процедура вирішення проблеми впорядкованою, скінченною, покроковою логікою та прямолінійним способом⁵. або "Процедура, що складається з кінцевого набору однозначних правил, які визначають кінцеву послідовність операцій, що забезпечує рішення задачі або певного класу проблем" - називається алгоритмом⁶. Алгоритми глибоко стурбовані процесом сортування та пошуку, що забезпечує ідеальну основу для застосування до пошуку інформації. Однак алгоритм пошуку шукає об'єкт або значення або елемент у структурі даних. Існує десятки алгоритмів, алгоритмічних прийомів, структур даних та підходів. Тут було зроблено спробу описати кілька важливих алгоритмів пошуку, які надзвичайно корисні та популярні для отримання цифрової інформації в бібліотеках. Ось такі:

- алгоритм Soundex;
- алгоритм метафона;

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

- алгоритм Фонекса;
- алгоритм стемінгу.

Алгоритм Soundex був спочатку розроблений Маргарет К. Оделл та Робертом С. Расселом. Вони запатентували оригінальний алгоритм Soundex у 1918 р. Як випливає з назви, цей метод базується на шести фонетичних класифікаціях звуків мовлення людини, які, в свою чергу, засновані на русі кліпів та язика для видачі цих звуків. Ці категорії можуть бути представлені як - двобічні (звук виробляється обома губами); Стоматологічний (шарнірно з'єднаний кінчиком або лезом язика проти або біля верхніх передніх зубів); Лабіоданти (Вимовляння за участю ковзання та зубів, лабіоданти звучать «f» та «v»); Альвеолярний (суглобовий з кінчиком язика, що торкається або знаходиться біля зубів); Велар (утворений задньою частиною язика, що торкається м'якого піднебіння або знаходиться поблизу нього, веларний „k“ з „вбивати“ прохолодний); Glottal (переривання потоку дихання під час мовлення через закриття голосової щілини). Технічно проблема існує щодо тих термінів, які часто неправильно написані. Навіть цій проблемі приділяється значна увага у зв'язку із системами бронювання авіакомпаній, системами каталогізації бібліотек, переписом населення, проектуванням баз даних та іншими програмами, що включають імена людей, коли є велика ймовірність того, що це ім'я буде помилково написано через поганий рукописний текст чи передачу голосу припустімо, вибір і тощо. Припустимо, що імена, що мають змінну довжину, можуть мати дивні написання або мати кілька написань, і іноді вони не є унікальними, особливо в різних культурах чи національних кордонах. Тому, щоб впоратися з цією постійною проблемою, слід розглянути фонетичні алгоритми, які можуть знайти подібні звучані терміни чи імена. Такі родини алгоритмів узагальнено називаються „Soundex”, за першою запатентованою версією. Мета полягає в тому, щоб перетворити аргумент у якийсь код, який має тенденцію поєднувати всі варіанти з однаковим ім'ям. обробка фонетичних варіацій (беззвучних приголосних) ппогана обробка імені, що містить частинки пРозрізняються системи транскрипції, хоча з алгоритмом soundex існує декілька проблем, але все ж це краще, ніж найкраще.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Основною проблемою для soundex та його похідних є те, що вони застосовуються як універсальний метод пошуку імен. Алгоритм, розроблений здебільшого для англійських імен, мало підходить для обробки імен із такими різноманітними звуковими моделями та структурами, як арабська, китайська, російська, і лише деякі з них.

Алгоритм метафону Тому оригінальний алгоритм Metaphone був розроблений як заміна алгоритму Soundex і був описаний у журналі ComputerLanguage. Він повертає приблизне наближення того, як звучить англійське слово, і використовує багато загальних правил англійської вимови, які SoundX не охоплює. Цей алгоритм зменшує слово (ключ) до коду символів від 1 до 4, використовуючи відносно прості фонетичні правила для типової розмовної англійської мови. Метафоне ігнорує всі випадки голосних після першої літери, але зберігаючи те саме, коли вони приходять як початкові літери слова.

Алгоритм Phonex Ранній успіх роботи (за допомогою Metaphone, моделі Германсена, моделі NYSIIS тощо) також порушує питання, чи можна методологію Вони порівняли ефективність декількох алгоритмів узгодження імен, включаючи основний метод soundex. Пошуки проводились за набором даних з 5600 унікальних прізвищ, обраних для подання імен, що починаються з кожної літери алфавіту, - частоти зустрічальності, що відображає фактичний алфавітний розподіл імен, включаючи також імена різної довжини. Дослідження показало, що soundex (який визнано найкращим із чотирьох порівняних алгоритмів) повернув лише 36,37% фактично правильних збігів, і що більше шістдесяти відсотків імен, які правильно відповідають іменам запитів, не повертаються. під час тестування, вони поступово змінювали код soundex, доки не було знайдено максимальну норму точних віддач, з мінімальним збільшенням неправильних збігів. Отриманий в результаті алгоритм отримав назву “Phonex”, що, мабуть, пропонує значні вдосконалення порівняно з soundex.

Алгоритм стемінгу Алгоритм стемінгу - це алгоритм, який перетворює слово у споріднену форму, тобто, що походить від кореневої форми слова. Одним з найпростіших таких перетворень є перетворення множини в однини,

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

іншим було б похідне дієслово від форми герундія (слово "-ing"). У процесі обробки природної мови (НЛП) стемінг - це процес злиття або об'єднання неідентичних слів, які відносяться до тієї ж основної концепції. Стермінг зазвичай здійснюється шляхом видалення будь-яких доданих суфіксів та префіксів з термінів індексу перед призначенням терміна. Загальновизнаним є те, що видалення закінчень слів (які часто називають зачищенням суфіксів) є хорошою ідеєю - ідея полягає в тому, щоб скоротити слова до їхніх коренів за допомогою автоматичної обробки закінчень слів. Префікси видалення також можуть бути корисними в деяких предметних доменах; хімія є очевидним прикладом (наприклад, діоксид, триоксид, цикло-бутен, цикло-пентен тощо). Оскільки стовбур терміна представляє більш широке поняття, ніж оригінальний термін, то процес стемлінгу випадково збільшує кількість отриманих документів (включаючи як корені слів, так і похідні слова) і потенційно покращує значення відкликання. Отже, походження слова є загальноприйнятою формою обробки мови у більшості систем пошуку інформації (IP). Це важлива функція, що підтримується сучасними системами індексації та пошуку.

Система пошуку репортерів (srs) разом із пошуковою системою пропонує, що коли ми шукаємо будь-яке ключове слово, то вона надає результат пошуку, точно пов'язаний із ключовими словами. srs разом із пошуковою системою дуже зручний та простий. користувач може легко шукати та отримувати інформативний результат із цієї системи. srs розроблений таким чином, що користувач ніколи не почуватиметься нудно з системою. Якщо користувач шукає ключове слово, тоді репортер пошуку завантажує результати пошуку з будь-якої пошукової системи, наприклад Google, а потім ранжирує сторінки на основі своєї інформації, що означає, що найбільш інформативна сторінка матиме рейтинг 1, тоді друга інформаційна сторінка отримає рейтинг 2 і так далі. Таким чином користувач отримує основну інформацію, яку він / вона насправді шукає. У цій системі є три кроки: шукайте будь-яке ключове слово в системі, отримуйте результати, як вони хочуть від Google, на основі рейтингу, а також відкривайте посилання та отримуйте найкращу інформацію.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

Використання алгоритму від google забезпечує чудовий веб-інтерфейс пошуку. Основні характеристики SRS наведені нижче:

- це економить значну кількість часу;
- він пропонує інформативний результат без аналізу всі результати, отримані від Google;
- це допомагає користувачам, які насправді не знають про пошук.

2.2 Цифрова бібліотека Bing Web Search API

Bing Search API - це тип когнітивних служб Azure, який забезпечує швидкий, простий і зручний пошук без реклами для кінцевих користувачів, отримуючи доступ до мільярдів веб-сторінок, зображень та відео за допомогою одного виклику API. Він класифікує результати на основі кількох факторів залежно від типу пошуку. Існує кілька типів Bing Search API, давайте розберемося в кожному з них детально.

Спеціальний пошук Bing забезпечує потужний та налаштовуваний досвід пошуку вмісту та тем, які нас цікавлять. Основна технологія працює у чотири кроки - визначає сайти та зображення, надає автоматичні пропозиції щодо запитів, застосовує рейтинг Bing і забезпечує відповідні результати. За допомогою цього API ми маємо повний контроль над результатами пошуку.

Іншими особливостями цього API є:

- покращує досвід пошуку завдяки автоматичним пропозиціям запитів;
- можливість блокувати конкретні запити з автоматичних пропозицій запитів;
- дозволяє налаштовувати заздалегідь визначені кольорові теми в розміщеному інтерфейсі.

Bing Entity Search дозволяє нашим програмам знаходити інформацію про такі організації, як люди, місця, бренди, місцеві компанії тощо на основі

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

пошукового запиту. Користувачі можуть фільтрувати результати за географічними межами, а також за категоріями.

Цей API дозволяє розробникам та користувачам проводити:

- пошукайте в інтернеті високоякісні статичні та анімовані зображення;
- налаштуйте пошук, щоб включити / виключити зображення на основі ринку, безпечного пошуку, свіжості, кольору, ліцензії та розміру;
- шукайте трендові зображення;
- відображення попереднього перегляду мініатюр;
- отримайте уявлення про зображення.

За допомогою цього API можемо шукати статті новин та актуальні теми з кількох категорій та з кількох джерел. Зазвичай результати включають ескіз статті, пов'язані новини та категорії, інформацію про постачальника, URL-адресу статті та дату публікації. Статті новин можна відфільтрувати на основі ринку, безпечного пошуку та свіжості.

Bing Video Search API спрощує додавання можливостей пошуку відео на веб-сайти та інші програми. Він шукає в Інтернеті відео та популярні відео. Ми можемо відфільтрувати результати за свіжістю, ціною, тривалістю відео та роздільною здатністю. Він надає корисні метадані, такі як творець, кількість переглядів, формат кодування тощо. Результати містять або статичну, або мініатюру руху.

Bing Visual Search дозволяє нам знаходити візуально подібні зображення зображень, знайдених в Інтернеті або завантажених зображень. Він може ідентифікувати штрих-коди та витягувати текстову інформацію із зображень, щоб надати уявлення. Цей API можна інтегрувати з іншими Bing API, такими як Bing Image Search API, Bing Entity Search API тощо.

Bing Web Search API використовує мільярди веб-документів для надання розширених деталей про пошукові запити. Це оптимально для програм, які потребують доступу до усього вмісту, що відповідає пошуковому запиту. Він також надає відповіді на обчислення, запити перетворення одиниць та запити, пов'язані з часом. Особливості цього API такі:

- результати з урахуванням місцезнаходження.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

- досвід пошуку без реклами.
- можливість встановити рівні безпечного пошуку.
- результати можна класифікувати на основі наших уподобань.
- надає пропозиції щодо написання та відповідні пошукові запити.

Перевірка орфографії Bing - це швидкий та ефективний API, який допомагає нам виявляти та виправляти граматику, розриви слів, сленг, омоніми та поширені помилки імен. Він також забезпечує підтримку нових брендів та інших вигаданих виразів. Він використовує машинне навчання та штучний інтелект для визначення контекстуальних та граматичних помилок.

Цей API надає користувачам інтелектуальні можливості попереднього введення тексту, тобто він повертає список запропонованих запитів на основі часткового рядка запиту у вікні пошуку, як тільки ми починаємо вводити текст. Його можна інтегрувати з іншими API Bing, такими як Bing Entity Search, Bing News Search тощо.

API пошуку Bing можна використовувати в багатьох сценаріях. Наприклад, ми можемо створювати набори даних, використовуючи Bing Image Search API, Bing News API тощо, і отримувати уявлення з цих наборів даних, виконуючи аналіз даних для покращення продуктивності організації. Іншим випадком використання може бути інтеграція Bing Spell Check API у веб-сайти або додатки, які займаються написанням вмісту для перевірки правопису та перевірки правопису.

В результаті проведеного дослідження було доведено, що обрана цифрова бібліотека у повній мірі надає можливості розробникам для створення повноцінного додатку для проведення онлайн перекладів тексту. Приклад використання можливостей бібліотеки наведено на рисунку 2.1

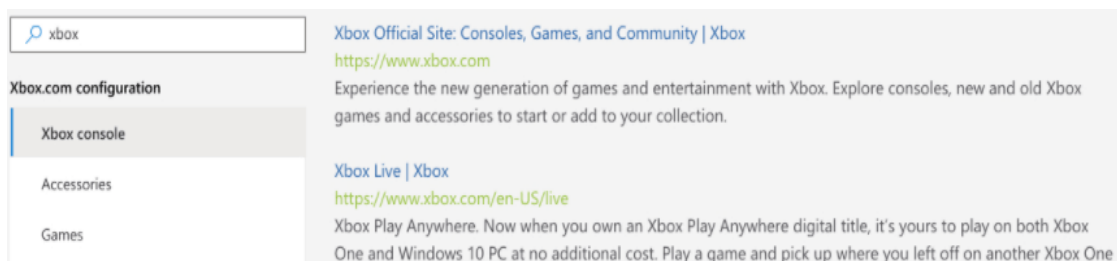


Рисунок 2.1 – Приклад використання Bing Web Search API

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

2.3 Алгоритм роботи програмного додатку онлайн пошуку інформації

Для виконання поставленого завдання, а саме для реалізації програмного додатку аналізу та пошуку інформації на основі запитів користувачів було спроектовано алгоритм на основі семантичного аналізу та використання цифрової бібліотеки Bing Web Search API. Основне навантаження на користувацьку частину програмного додатку полягає в чіткому виділенні головних слів та ранжування їх по мірі аналізу нових запитів від користувача. Пошук інформації здійснюється на серверній частині, що дозволяє зменшити навантаження локальну робочу стацію, а також на канал зв'язку. Тому що користувацька частина програмного додатку буде виступати в основному як інтерфейс між користувачем та серверною частиною програмної розробки. Час роботи програми напряму залежить від швидкості каналу зв'язку між користувачем та сервером, а також від рівня складності заданого запиту. Запит може містити слова, що рідко поєднуються між собою у звичайній мові, наприклад «сніг в пустелі», а отже і записів з поєднанням таких слів буде небагато, отже сервер поверне мало сторінок, а отже і відповідь прийде з невеликою затримкою. У протилежній ситуації, кількість співпадінь буде великою, що автоматично збільшує кількість відповідей від сервера. Проте в загальному, часові втрати при проведенні є відносно незначними та дозволяють значно зменшити час при подальшій обробці інформації.

Запропонований алгоритм містить наступні етапи роботи:

Етап 1. Початок роботи з програмою (вибір мови, перевірка доступу до глобальної мережі та серверу програми).

Етап 2. Проведення перевірки наявності та правильної роботи усіх компонентів програмної системи, при необхідності завантажити необхідні компоненти.

Етап 3. Очікування від користувача текстового запиту.

Етап 4. Розбиття користувацького запиту на окремі слова та видалення стоп-слів.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Етап 5. Видалення слів що повторюються та сортування їх в порядку повторної появи. Якщо слово з'явилося перший раз то воно заноситься в базу даних і йому присвоюється рівень 1, якщо слово вже колись було використано в запитах, то отримуємо з бази даних його мітку збільшуємо на 1 та запам'ятовуємо нову мітку.

Етап 6. Проводиться передача відкоригованого запиту користувача до серверної частини програмного додатку.

Етап 7. Очікування відповіді від сервера, якщо відповідь не прийшла з час $T=20$ секунд, то проводить перевірка стану з'єднання між сервером та програмним додатком. Якщо все гаразд то проводиться повторна подача запиту, інакше вивід повідомлення про помилку та перехід на крок 9.

Етап 8. Отримання результатів обробки запиту та їх ранжування. Ранжування відбувається відносно кількості повторів у відповіді слів, що знаходились в запиті.

Етап 9. Вивід отриманих результатів.

В спроектованому алгоритмі містяться усі етапи обробки та аналізу текстових запитів від користувачів та збережені основні етапи програм – аналогів. Алгоритм представлений у вигляді блок-схеми та дозволяє користувачам швидко та в інтуїтивно зрозумілій формі побачити необхідний алгоритм. При цьому слід зазначити, що в алгоритмі відсутня перевірка орфографічної правильності введеної користувачем інформації, даний етап міг би значно підвищити рівень якості введених користувачем запитів, проте з іншої сторони збільшив би час обробки. При цьому слід зазначити що дана перевірка здійснюється на серверній частині. Графічно запропонований алгоритм можна представити у вигляді наступної блок-схеми (рисунк 2.2):

До переваг розробленого підходу такі:

- проста реалізація та висока швидкість роботи;
- принципи формування запитів однозначно дозволяють виділити слова, що частіше за інші вводяться користувачем;
- алгоритм не є вимоливим до ресурсів робочої станції;
- висока швидкість опанування навиками роботи з алгоритмом.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

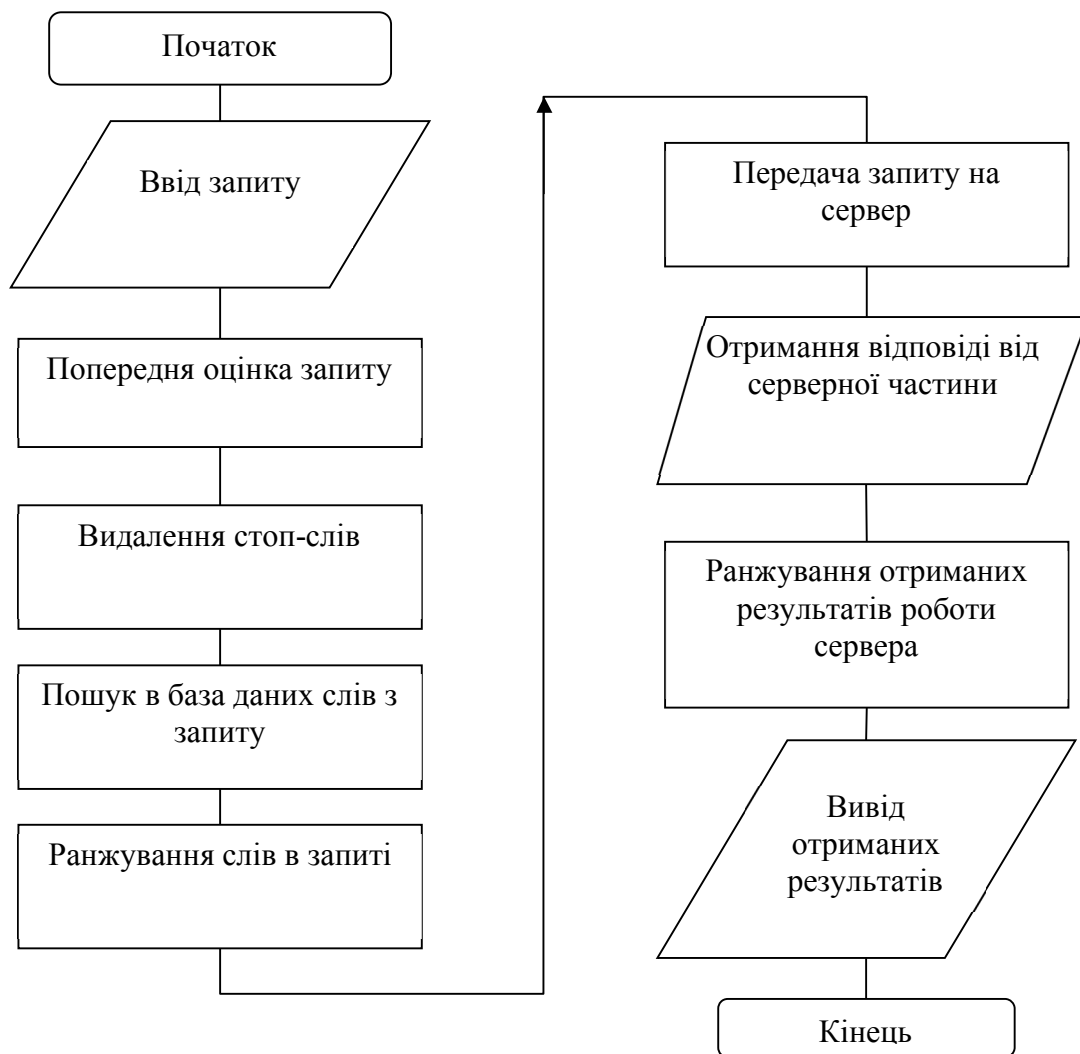


Рисунок 2.3 – Блок-схема алгоритму додавання нового елемента до блок-схеми алгоритму

Серед недоліків є:

- при втраті значень з вагами слів алгоритм буде навчатись по новій, що може на початкових етапах надавати не зовсім коректні результати;
- якщо користувацький запит містить багато орфографічних помилок, то результат відповіді сервера можуть бути не передбачуваними.

3. ПРОГРАМНА СИСТЕМА ОНЛАЙН ПОШУКУ ІНФОРМЦІЇ НА ОСНОВІ ТЕКСТОВИХ ЗАПИТІВ

3.1 Структура програмної системи пошуку інформації

Для реалізації програмного забезпечення та проведення подальших тестів запропонованих алгоритмів було обрано мову програмування java. Даний програмний продукт є широко поширеним серед розробників як прикладних програмних продуктів так і мережових. Java - мова загального призначення, надійна, безпечна та об'єктно-орієнтована мова програмування. Це мова високого рівня, тобто її синтаксис використовує англійську мову, як мова. Він був розроблений Sun Microsystems у 1995 році. Зараз він підтримується та розповсюджується Oracle. Java має своє середовище виконання та API; тому його ще називають платформою. Протягом багатьох років Java використовується у великій кількості програм. Однак він має різні переваги та недоліки, наведені нижче. Переваги:

Простота. Java - це проста мова програмування, оскільки її легко вивчити та зрозуміти. Його синтаксис заснований на C ++, і він використовує автоматичний збір сміття; тому нам не потрібно видаляти об'єкти без посилань із пам'яті. Java також видалила такі функції, як явні вказівники, перевантаження оператора тощо, що полегшує читання та запис.

Об'єктно-орієнтованість. Java використовує об'єктно-орієнтовану парадигму, що робить її більш практичною. Все в Java - це об'єкт, який опікується як даними, так і поведінкою. Java використовує об'єктно-орієнтовані концепції, такі як об'єкт, клас, успадкування, інкапсуляція, поліморфізм та абстракція.

Захищеність. Java є захищеною мовою програмування, оскільки вона не використовує явні вказівники. Крім того, програми Java працюють усередині пісочниці віртуальної машини. JRE також надає завантажувач класів, який використовується для динамічного завантаження класу в JVM. Він відокремлює пакети класів локальної файлової системи від тих, які імпортуються з мережі.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Надійність. Java - це надійна мова програмування, оскільки вона використовує потужне управління пам'яттю. Розробники також можуть обробляти винятки за допомогою коду Java. Крім того, можемо використовувати перевірку типу, щоб зробити наш код більш безпечним. Він не надає явних покажчиків, щоб програміст не міг отримати доступ до пам'яті безпосередньо з коду.

Незалежна платформа. Код Java може працювати безпосередньо на декількох платформах, тобто нам не потрібно кожного разу компілювати його. Це правильно один раз, працює в будь-якій мові (WORA), яка може бути перетворена в байт-код під час компіляції. Байт-код - це незалежний від платформи код, який може працювати на декількох платформах.

Багатопоточність. Java використовує багатопотокове середовище, в якому більші завдання можна перетворити на різні потоки і запустити окремо. Головною перевагою багатопоточності є те, що нам не потрібно надавати пам'ять кожному поточному потоку.

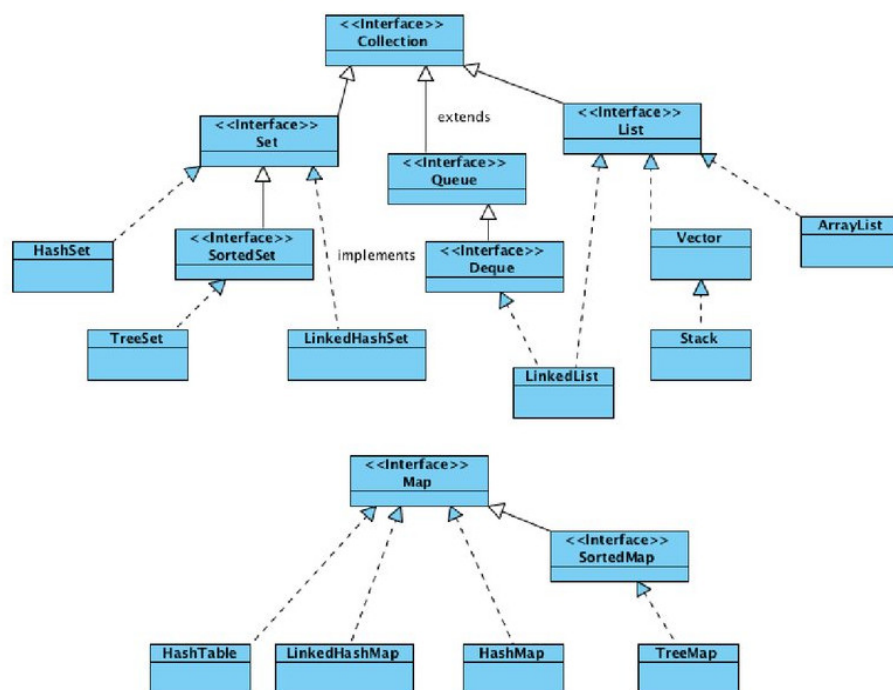


Рисунок 3.1 – Приклад ієрархії класів в java

Саме ці переваги мови програмування дозволило реалізувати запропоновані алгоритми та провести порівняльний аналіз.

При проектування програмного додатку значний час було приділено розробці його структури. Оскільки програмний додаток в своїй основі базується на технології «клієнт – сервер», то значну увагу слід було приділити організації коректного інтерфейсу між двома частинами. Тому запропонована архітектура є блочною. Саме такий підхід дозволить в подальшому проводити удосконалення та модернізацію окремих програмних функцій без руйнування цілісності цілої системи. Оскільки при заміні окремого модуля відбувається корекція роботи лише окремої частини функцій. При цьому загальний підхід та архітектура будуть працювати в стабільному режимі з дотриманням принципів та правил передачі інформації між окремими блоками.

Клієнт-серверна модель описує, як сервер надає ресурси та послуги одному або декільком клієнтам. Приклади серверів включають веб-сервери, поштові сервери та файлові сервери. Кожен із цих серверів забезпечує ресурси для клієнтських пристроїв, таких як настільні комп'ютери, ноутбуки, планшети та смартфони. Більшість серверів мають взаємозв'язок "один до багатьох" з клієнтами, тобто один сервер може одночасно надавати ресурси кільком клієнтам (рисунок 3.2).

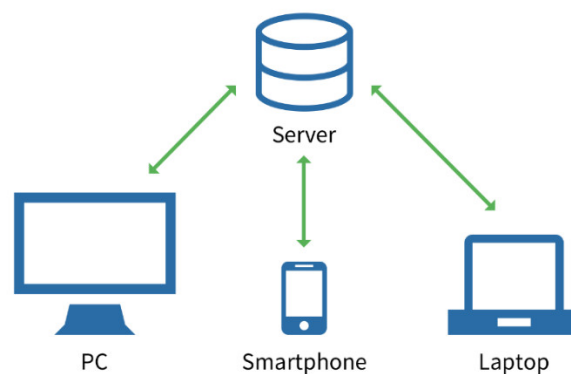


Рисунок 3.2 – Принцип «клієнт-серверної» моделі

На основі аналізу системних вимог встановлено набір усіх функцій, які програма повинна підтримувати. Потім отримані функції об'єднують у логічно пов'язані групи. Кожна з цих груп може бути складовою частиною програмної системи.

Запропоноване архітектурне рішення включає в себе усі не обідні структурні одиниці для повноцінного функціонування програмної системи. Серед основних функцій - семантичний аналіз отриманого повідомленн, робота з базою дани коефіцієнтів, а також параметри передачі та отримання зпитів між клієнтською та серверними частинами.

Загальна архітектура програмного забезпечення представлена у вигляді структури, що ілюструє окремі функційональні блоки різних частин цілісної програмної системи. Стуруктура наведена на рисунку 3.3.

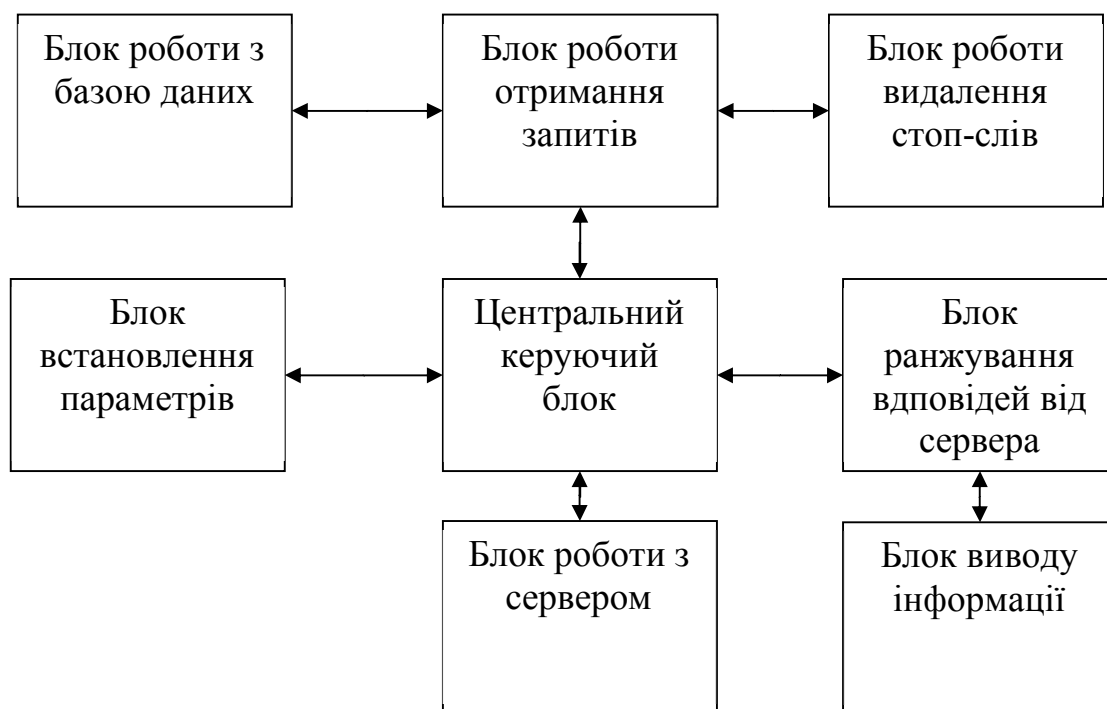


Рисунок 3.3 – Узагальнена структура програмного додатку пошуку інформації на основі текстових запитів

При аналізі структури програмного додатку детальніше зупинемось на основних програмних модулях:

– “Блок встановлення параметрів” – набір функцій та процедур, що надають користувачеві можливості регулювати налаштування роботи програми або змінювати зовнішній вигляд графічного інтерфейсу користувача. Інші функції включають налаштування параметрів системи відображення інформації

для користувача (колір, товщина лінії, налаштування шрифту тощо). Цей пристрій підтримує роботу системи проте не є головним.

– “Блок взаємодії з сервером” – даний архітектурний модуль призначений для організації обміну повідомленнями між клієнською та серверною частинами програмного додатку, серед його основних завдань є перевірка наявності з’єднання між двома частинами, повторного перепідключання у випадку втрати зв’язку, форматування запиту до сервера відносно встановлених принципів та правил організації взаємодії. Даний сервер є один з двох найважливіших елементів системи;

– “Блок отримання запитів” – функції даного блоку реалізують можливість користувача ввести у відповідному полі запит у вигляді текстового повідомлення або набуту слів, на основі якого в подальшому буде сформований запит для серверної частини. Даний модуль не є основним, проте без його коректної роботи в системі не було б можливості отримувати від користувача інформації.

– “Блок роботи з базою даних” – в роботі використовується реляційна база даних у вигляді таблиці, в якій попарно зберігають окремі слова та коефіцієнти частоти їх появи у відповідних запитах користувача. Даний модуль проводить пошуку відповідних ваг за значенням слова. При цьому в системі передбачено очищення бази даних від мало використовуваних слів, це відбувається шляхом виділення малоповторюваних слів. Частоту перевірок та мінімальну частоту зустрічань користувач може встановити в ручному режимі.

– “Блок роботи з базою даних” – призначення функцій даного блоку це проведення попередньої обробки вхідного повідомлення від користувача та видалення з нього всіх слів, що є в списку стоп-слів. Це сприяє зменшенню довжини стрічки та збільшенню інформативності відповідного запиту.

– “Центральний керуючий блок” – головний елемент архітектури програмного додатку, означає основний компонент дизайну програми, який з’єднується з окремими елементами програми. Інші особливості, властиві цій збірці, включають можливість перевірки цілісності та дійсності програми. Окрім того в ньому закладені принципи для перевірки правильності та цілісності

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

передачі даних між окремими функціональними блоками, що в разі піднімає стійкісні характеристики роботи програмної системи в цілому.

Відповідальність за кожен компонент походить від запропонованої архітектури. Компонент відповідальний, а зв'язок між окремими компонентами зведений до мінімуму. Архітектура визначає правила спілкування між пунктами порядку денного та описує, які компоненти можна використовувати безпосередньо, а які не слід використовувати безпосередньо та повністю. Ієрархічна модель програмної системи якісно відрізняє її від інших модклей, оскільки на її основі легко можна нарощувати функціональні моливості програмної розробки в цілому.

Для проведення попереднього моделювання розроблювального програмного додатку, використовуються ряд UML діаграм. Вибір UML пов'язаний з тим, що:

Ваша програмна система професійно розроблена і задокументована перед кодуванням. Ви будете точно знати, що отримуєте, заздалегідь.

Оскільки дизайн системи стоїть на першому місці, багаторазовий код легко помітити та закодувати з найвищою ефективністю. Ви матимете менші витрати на розробку.

Логічні «діри» можна помітити на кресленнях дизайну. Ваше програмне забезпечення буде поводитися так, як ви очікуєте. Сюрпризів стає менше.

Загальний дизайн системи визначатиме спосіб розробки програмного забезпечення. Правильні рішення приймаються до того, як в реальності стикнетесь з погано написаним кодексом. Знову ж кошторис загальних витрат буде меншим.

UML дозволяє побачити загальну картину. Розробники можуть розробити більше ефективних архитектур та системних рішень.

Коли архітектори повертаються, щоб внести зміни у систему, набагато простіше працювати в системі, яка має документацію UML. Набагато менше відбувається «перенавчання». Ваші витрати на обслуговування системи будуть нижчими.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо вам знадобиться працювати з іншим розробником, діаграми UML дозволять їм швидко пришвидшити роботу у вашій користувацькій системі.

Якщо розробникам потрібно спілкуватися із зовнішніми підрядниками або навіть із власними програмістами, це набагато ефективніше.

Легко налагоджувати будь-яку проблему. Більш обширна програма може зайняти до годин і годин пошуку, щоб знайти помилку в ній, а також може спричинити проблеми в системі пізніше. Отже, програма добре розроблена за допомогою UML, так що кожне із завдань містить свої власні коди, які програмісту легше налагоджувати.

В першому варіанті розгдатиметься розроблювальна система як сукупність акторів та прецедентів з якими вони можуть взаємодіяти під час роботи з програмою (рисунок 3.4):

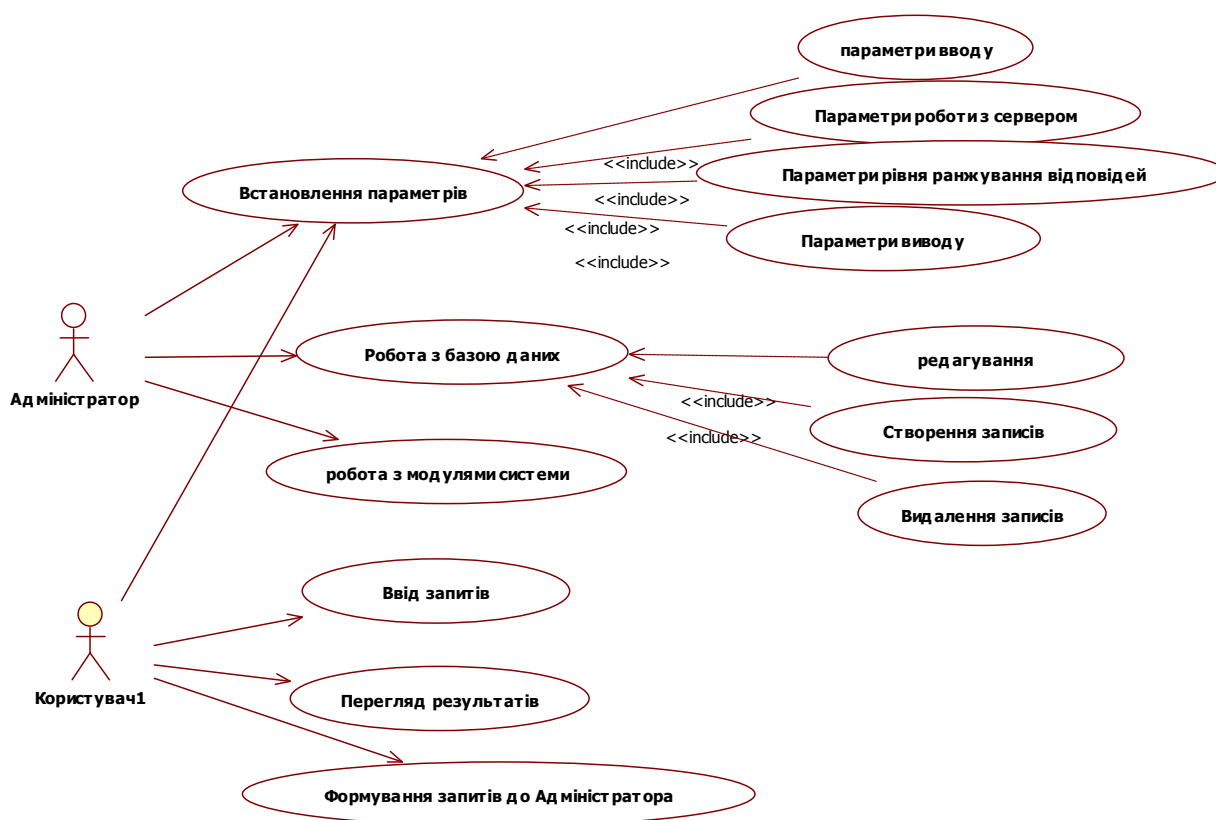


Рисунок 3.4 – Діаграма прецедентів програмного додатку опису пошуку інформації на основі текстових запитів користувачів

При моделюванні було виділено два типи акторів, що будуть взаємодіяти з програмною системою. Перша група акторів – це “Адміністратор”. Дана група не стільки буде користуватись програмним додатком скільки буде встановлювати параметри роботи окремих модулів, слідкувати з коректністю роботи та своєчасним оновленням та навчання нових користувачів принципам роботи з системою. Актори типу "Користувач" мають змогу створювати, редагувати, відправляти запити на серверну частину програмного засобу та переглядати отримані результати. При чому обидва цих класи акторів можуть встановлювати загальні параметри роботи програм.

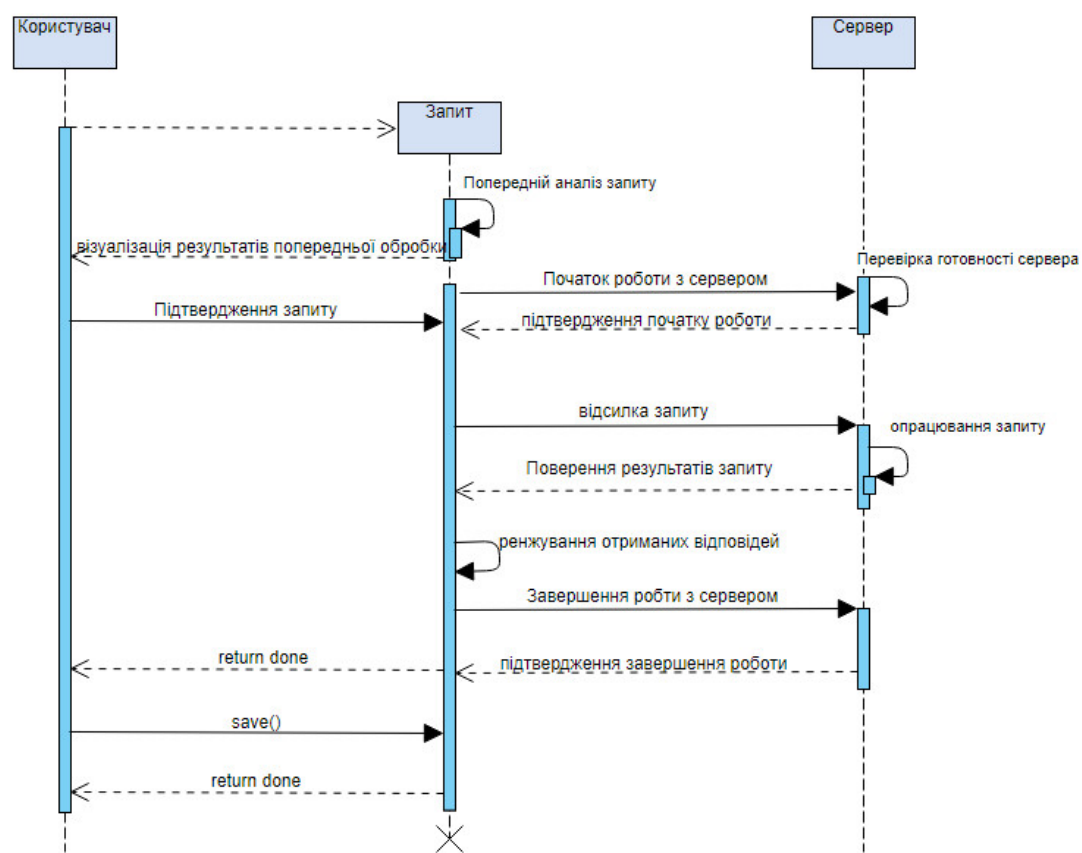


Рисунок 3.5 – Діаграма послідовності взаємодії користувачів з програмною системою

Моделювання послідовності кроків роботи з програмною системою проілюстрував, що в програмі:

- відсутні неоднозначні дії, що можуть сповільнити програму;
- для досягнення мети користувачу слід виконати мінімальну кількість дій;

— час на освоєння програми є мінімальний та не залежить від знань комп'ютерної грамотності.

Виділення двох типів акторів які мають певний набір функцій, при роботі з програмною системою є цілком достатнім вирішення завдань користувачів при цьому робота програмної розробки буде проходити без залучення значних апаратних ресурсів. Час реакції на дії користувачів буде мінімальним, а результат своїх дій користувач бачитиме відразу.

Графічний інтерфейс користувача був розроблений на етапі аналізу вимог. Архітектура описує основні аспекти графічного інтерфейсу користувача (GUI). Використання графічного інтерфейсу користувача може врешті-решт призвести до популярності або відмови програмного продукту. При створенні графічного інтерфейсу користувача було вирішено розділити головне вікно на окремі практичні області. Це розділення зменшило візуальне навантаження на користувача та скоротило час користувачів з меншим досвідом та навичками для оволодіння програмою з комп'ютерами та графічним інтерфейсом операційних систем. Результати проектування та створення графічного інтерфейсу користувача показано на рисунку 3.6.

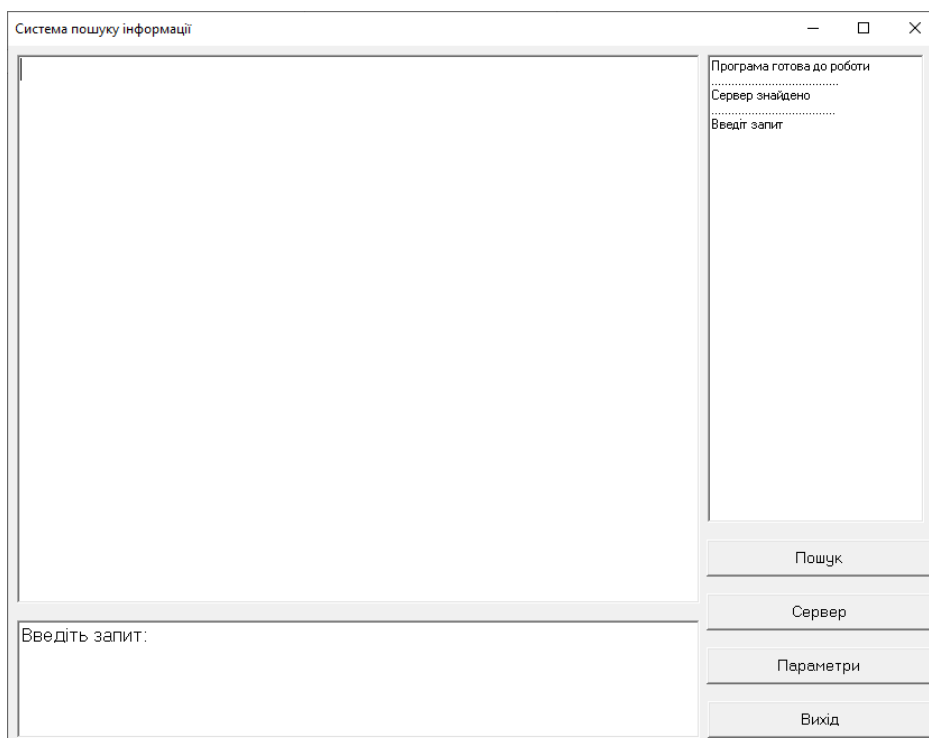


Рисунок 3.6 – Загальний вигляд головного вікна програмного додатку

Серед основних структурних елементів графічного інтерфейсу слід відмітити:

– Основна робоча область програмного додатку. Дана область має найбільшу площу, оскільки це пояснюється тим. Що саме там будуть відображатись майбутні результати пошуків інформації за запитом. Відповідно основну частину часу при роботі з системою користувач буде проводити саме там. Іншою частиною даного блоку є невеличке вікно, де користувач може ввести запит у вигляді текстового повідомлення. Дана область розміщується прямо під головним вікном, що дозволяє користувачеві втрачати мінімальний час для того щоб переключити свою увагу з однієї частини на іншу.

– Вікно виводу системної інформації – ще одна досить велика частина робочого вікна програмного додатку, що забезпечує інформування користувачів про перебіг процесів в середині програми. Дана інформація є більше корисною для користувачів, які матимуть більший досвід роботи з програмною, оскільки дозволяє передавати стан виконання конкретної задачі в режимі реального часу. Для адміністраторів дана інформація носить суттєвий характер, оскільки дозволяє вносити зміни в роботу програмного додатку.

– Керуючий блок перемикачів – даний блок реалізований у вигляді набору активних кнопок, що активують ряд відповідних функцій. Серед яких активація процесу пошуку, активація меню для встановлення параметрів роботи з серверною частиною програмного додатку, а також активація підменю для встановлення параметрів роботи локального додатку програмної системи.

Архітектура програми є модульною, щоб графічний інтерфейс можна було змінити, не зачіпаючи основну логіку програми. Використання даного принципу гарантує, що при необхідності внесення змін в програмний додаток, це відбудеться без значних часових затрат. Додатково ж програмний модуль реалізований у стилі програмних додатків операційної системи Windows, тому нові користувачі, що мали досвід роботи з даною операційною зможуть швидко її опанувати. Ще однією безумовною перевагою є мінімальна кількість візуальних елементів, що дозволить користувачам не відволікатись від роботи з самою системою.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Підсистеми онлайн пошуку інформації

Під час реалізації програмного додатку були виокристані основні можливості мови програмування java для проектування та реалізації програмних додатків. Зокрема був реалізований основний клас в якому було спроектовано метди для коректної візуалізації та роботи з веб-сторінками.

```
public Browser() {  
    super("Система пошуку інформації");  
    setSize(500, 300);  
    getContentPane().setLayout(new BorderLayout());  
  
    JPanel p = new JPanel();  
    p.setLayout(new BoxLayout(p, BoxLayout.X_AXIS));  
    p.add(new JLabel("Address"));
```

Зокрема наведено код для встановлення параметрів робти системи на старті. Причому після початкового етапу робти програми користувачі можуть встановити власні , більш зручні для себе параметри роботи.

```
m_locator.load("addresses.dat");  
BrowserListener lst = new BrowserListener();  
m_locator.addActionListener(lst);  
MemComboAgent agent = new MemComboAgent(m_locator);  
p.add(m_locator);  
getContentPane().add(p, BorderLayout.NORTH);
```

Після отримання відповідей від ерверної частини програмний додаток пробує відобразити отриманий список веб-документів на головному вікні програми. Відображення могу відбуватись як у вигляді списку знайдених документів, так і у вигляді відповідного сайту. Дана моливість забезпечується можливостями роботи класів java з елементами веб-сторінок.

```
m_browser = new JEditorPane();  
m_browser.setEditable(false);  
m_browser.addHyperlinkListener(lst);  
JScrollPane sp = new JScrollPane();  
sp.getViewport().add(m_browser);  
getContentPane().add(sp, BorderLayout.CENTER);
```

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

WindowListener wndCloser = new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        m_locator.save("addresses.dat");
        System.exit(0);
    }
};
addWindowListener(wndCloser);
setVisible(true);
m_locator.grabFocus();
}

```

Перемикання між окремими елементами списку сайтів здійснюється з основі класу `BrowserListener`. Даний клас було реалізовано для отримання можливості працювати з списком який прийшов не від сервера напряму, а був вже переконвертований під власні потреби програмного додатку.

```

class BrowserListener implements ActionListener,
HyperlinkListener {
    public void actionPerformed(ActionEvent evt) {
        String sUrl = (String) m_locator.getSelectedItemAt();
        if (sUrl == null || sUrl.length() == 0 )
            return;
        BrowserLoader loader = new BrowserLoader(sUrl);
        loader.start();
    }
}

```

Окрім того в даному класі реалізовано механізм перевірки активності наданої під час пошуку веб-адреси. Цей механізм був передбачений для того щоб у випадку відсутності зв'язку з віддаленим сервером на якому розміщено відповідний сайт програма не витратила зайві зусилля на його завантаження.

```

public void hyperlinkUpdate(HyperlinkEvent e) {
    URL url = e.getURL();
    if (url == null )
        return;
    BrowserLoader loader = new BrowserLoader(url.toString());
    loader.start();
}

```

Дані класи та методи, що були реалізовані під час роботи на виконанням послевленої мети у повній мірі забезпечують виконання поставлених завдань та дозволяють реалізувати усі запропоновані алгоритми для аналізу та пошуку інформації на основі текстових запитів користувачів

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Тестування модуля пошуку інформації на основі текстових запитів

Технічні характеристики робочої станції є достатніми для проведення тестування розробленого програмного додатку та отримання достовірних даних про коректність роботи, швидкість опрацювання зображень та можливість подальшого впровадження запропонованої програмної розробки.

Для коректної оцінки результатів тестування вещь процес було вирішено проводити в декілька етапів та з використання запитів різної складності та структури. Причому як структура запиту (від простого набору непов'язаних між собою слів до чітких та звичних нам словосполучень) до кількості слів в запиті (кількість слів в запиті змінювалась від одного-двох до майже 10 і це без врахування стоп-слів). Результат роботи з простим запитом, що складається з 3 логічно взаємопов'язаних слів наведено на рисунку 3.7.

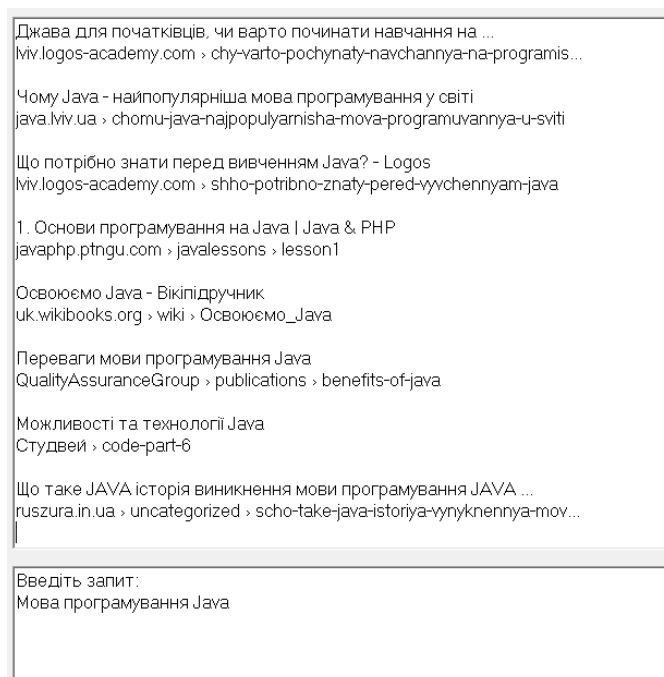


Рисунок 3.7 – Приклад роботи програмного додатку

Як видно з отриманих запитів, всі вони логічно підходять до введенного запиту та містять усі слова які були передані під час тестування. Кщо

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

проаналізувати кількісні результати проведеного дослідження то можна розглянути дані, що були показані у системному вікні (рисунок 3.8).

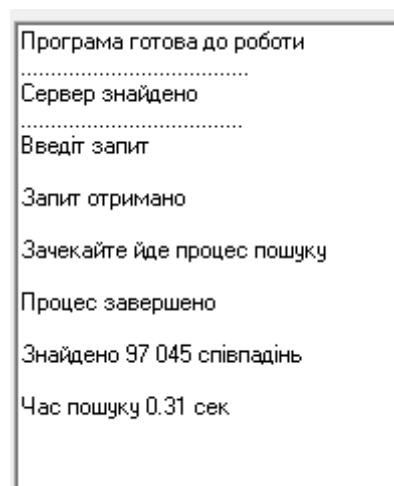


Рисунок 3.8 – Приклад роботи програмного додатку

Як видно з отриманої від сервера відповіді, за результатами пошуку було отримано більше 97 тисяч відповідей і список даних сайтів було передано у програмний додаток, де користувач матиме змогу всі їх переглянути. Час спрацювання програмного додатку є меншим ніж 1 секунда, що в принципі надає право стверджувати, що програмний додаток працює в режимі реального часу та може бути використаний для швидкого пошуку інформації. Звісно на швидкість впливали і зовнішні фактори такі як швидкість каналу зв'язку та високий рівень апаратних ресурсів робочої станції. Проте без швидких алгоритмів та правильно підібраних зовнішніх цифрових бібліотек результат роботи був би на порядок нижчим.

Під час останнього етапу тестування було перевірено додаткові функції програмного додатку, а саме:

- можливість модифікації параметрів роботи програмного додатку;
- можливості масштабування робочого проекту;
- функції автоматичного відкриття знайдених веб-сайтів на головному вікні проекту;
- перевірка коректного функціонування контекстного та випадаючого меню.

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

Метою техніко – економічного розділу кваліфікаційної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності програмного додатку згідно затвердженого технічного завдання та на основі обраних методів, алгоритмів та технологій, а також прийняття рішення про його подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки. Для проведення даного дослідження необхідно провести ряд розрахунків.

4.1 Розрахунок витрат на розробку програмного додатку

Витрати на розробку і впровадження програмного додатку (K) на основі запропонованих алгоритмів та розробленої структури, що враховують результати аналізу програм-аналогів включають:

$$K = K_1 + K_2,$$

де K_1 – витрати на розробку апаратного та програмного забезпечення грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку апаратних та програмних засобів включають:

- витрати на оплату праці розробників ($B_{оп}$);
- витрати на відрахування у спеціальні державні фонди (B_{ϕ});
- витрати на матеріали та комплектуючі ($П_{\epsilon}$);
- накладні витрати (H);
- інші витрати (I_{ϵ})
- витрати на використання комп'ютерної техніки ($B_{КТ}$) .

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Розрахунок витрат на оплату праці.

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоємності відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломант; консультант техніко-економічного розділу (таблиця 4.1).

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

№п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, ст. викладач	7449
2	Консультант техніко-економічного розділу, ст. викладач	7449
3	Студент	1400

Витрати на оплату праці розробників проекту визначаються за наступною формулою (4.1):

$$B_{OP} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} , \quad (4.1)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду.

Середньо годинна ставка працівника може бути розрахована за такою формулою (4.2):

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PQ_i}, \quad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

PQ_i - місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Коефіцієнт h , який визначає розмір додаткової заробітної плати, для керівника та консультанта техніко-економічного розділу дорівнює 0,47.

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, старший викладач	16	64,3	1028,8
2	Консультант техніко-економічного розділу, доцент	2	59,9	119,8
3	Студент	144	8,33	1199,52
Разом				2348,12

Відрахування на соціальні заходи. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати:

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

$$B_{\phi} = \frac{20,5}{100} \cdot 2348,12 = 481,36 \text{ грн.}$$

Розрахунок витрат на матеріали та комплектуючі.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою (4.3):

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

де K_i – витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i – ціна за одиницю i -го типу матеріалу, грн.;

i – тип матеріального ресурсу;

n – кількість типів матеріальних ресурсів.

Таблиця 4.3 – Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. вимі ру	Факт. витраче но матеріа лів	Ціна за одиниц ю, грн.	Сума, грн	Транспо ртні витрати (10% від суми)	Загальна сума, грн
	Допоміжна література	шт	1	1000	1000	100	1100
	Папір (формат А4)	уп	2	100	200	20	220
	Ручка кулькова	шт	2	10	20	2	22
	Олівець простий	шт	2	10	20	2	22
	Диски CD-R	шт	2	20	40	4	44
	Зошит, 96 арк	шт	1	50	50	5	55
	Тонер для принтера	уп	1	90	90	9	99
	Канцелярські маркери (синій, зелений)	шт	2	20	40	4	44
	Р а з о м						1606,00

Витрати на використання комп'ютерної техніки.

Витрати на використання комп'ютерної техніки (B_{KT}) включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ЗУНУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 12 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

№ п/ п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера грн.
1	Проведення досліджень та оформлення їх результатів у вигляді звіту	60	7200
2	Оформлення техніко-економічного розділу	8	96
3	Оформлення ДП	12	144
Разом		80	960

Накладні витрати.

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 2348,12 = 3522,18 \text{ (грн).}$$

Інші витрати.

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I_B = 2348,12 \cdot 0,1 = 234,81 \text{ (грн)}.$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_M + H + I_B + B_{KT},$$

$$K_1 = 2348,12 + 481,36 + 1606,00 + 3522,18 + 234,81 + 960,00 = 8849,47 \text{ (грн)}.$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою (4.4):

$$K_2 = S_{м.г.} \cdot t_{від} \quad (4.4)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн./год;

$t_{від}$ – комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 12 грн., тому $K_2 = 12 \cdot 60 = 720$ грн.

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблювальної програмної системи слід порівняти її з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi},$$

де E_{Π} – одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ – вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

$E_{2\Pi}$ – вартість роботи комп'ютера для виконання проектного рішення (аналог), грн.

Річні експлуатаційні витрати $B_{E\Pi}$ визначаються за формулою:

$$B_{E\Pi} = E_{\Pi} * N_{\Pi},$$

де N_{Π} – періодичність експлуатації ПЗ (аналог), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1\Pi} = \sum_{i=1}^n n_i t_i c_i,$$

де i – категорії працівників, які приймають участь у підготовці відповідних даних ($i=1,2,...n$);

n_i – кількість працівників i -ої категорії, осіб.;

t_i – трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i – середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1 + b)}{m},$$

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

де c_i^0 – основна місячна заробітна плата працівника i -ої категорії, грн.;

b – коефіцієнт, який враховує додаткову заробітну плату (прийmemo 0,57);

m – кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c = 6061$ грн.

Тоді:

$$c_1 = \frac{6061(1 + 0,57)}{22 * 8} = 56,64 \text{ грн/год}$$

Трудовістість підготовки даних для проектного рішення складає 1 год., для аналога 1,5 год.

Таблиця 4.5 – Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати , грн.
	Проектне рішення		
1	1	56,64	56,64
	Аналог		
1	1,5	56,64	84,95

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2\Pi} = t * S_{MG}$$

Де t – витрати машинного часу для реалізації рішення (аналогу), год.;

S_{MG} – вартість однієї години роботи комп'ютера, грн./год.

$$E_{2\Pi} = 1 * 12 = 12 \text{ грн.}; E_{2A} = 1,5 * 12 = 18 \text{ грн.}$$

$$E_{\Pi} = 56,64 + 12 = 68,64 \text{ грн.}; E_A = 84,95 + 12 = 96,95 \text{ грн.}$$

$$B_{E\Pi} = 68,64 * 252 = 17297,28 \text{ грн.}; B_{EA} = 96,95 * 252 = 24431,4 \text{ грн.}$$

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

Обчислення накладних витрат.

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників.

$$H_B = 0,7 * B_{OP} = 0,7 * (c_1 * 168), \quad (4.7)$$

де H_B – накладні витрати.

$$H_B = 0,7 * 9515,52 = 6660,86 \text{ грн.}$$

Складання кошторису витрат та визначення собівартості. Результати проведених розрахунків зведемо у таблицю 4.6.

Таблиця 4.6 – Кошторис витрат ($B_{КС}$)

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці (B_{OP})	2348,12
2	Відрахування у спеціальні державні фонди (B_{Φ})	481,36
3	Витрати на матеріали та комплектуючі (B_M)	1606,00
4	Накладні витрати на розробку (H)	3522,18
5	Інші витрати (I_B)	234,81
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту (K_2)	720
7	Накладні витрати експлуатацію (H_B)	6660,86
8	Річні експлуатаційні витрати (B_{EA})	24431,4
Разом		40004,73

Розрахунок ціни проекту.

Договірна ціна ($Ц_D$) для проектних рішень розраховується за формулою (4.8):

$$Ц_D = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.8)$$

де $B_{КС}$ – кошторисна вартість, грн.;

p – середній рівень рентабельності, % (приймаємо 20% за погодженням з керівником).

$$Ц_D = 40004,73 \cdot (1 + 0,2) = 48005,68 \text{ грн.}$$

4.3 Визначення економічної ефективності та терміну окупності

Економічна ефективність (E_ϕ) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_\phi = \frac{\Pi}{B_{КС}}, \quad (4.9)$$

де $\Pi = Ц_D - B_{КС}$ – прибуток, грн.;

$B_{КС}$ – кошторисна вартість, грн..

$$E_\phi = 8000,95 \text{ грн.} / 40004,73 \text{ грн.} = 0,2.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (Tr):

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

$$T_P = \frac{1}{E_P} . \quad (4.10)$$

Тобто: $T_P = 1/0,2 = 5р.$

Прийнятним вважається термін окупності близький до 7 років.

Розраховані економічні показники проекту занесемо до таблиці 4.7.

Таблиця 4.7 – Економічні показники розробки

№ п/п	Показник	Значення
1.	Собівартість, грн.	40004,73
2.	Плановий прибуток, грн.	8000,95
3.	Ціна, грн.	48005,68
4.	Економічна ефективність	0,2
5.	Термін окупності, рік	5

Враховуючи основні економічні показники з таблиці 4.7, можна зробити висновок, що при економічній ефективності 0,2 та терміні окупності – 5 роки проводити роботи по впровадженню даного програмного додатку є доцільним та економічно вигідним.

ВИСНОВКИ

На основі аналізу сучасних алгоритмів та технологій аналізу дани з використанням семантичного аналізу, методів «клієнт-серверної» технології та проведених експериментів, можна зробити наступні висновки:

1) Досліджено та проведено класифікацію різних груп інформації, на основі існуючих стандартів, що надало можливість виділити основні характеристики інформації.

2) Проаналізовано можливості web-технологій для пошуку та аналізу інформації, що дозволило виділити стек технологій на основі яких будувалась структура програмного модуля пошуку інформації.

3) Проведено аналіз сучасних програмних онланй систем пошуку інформації, що дозволило виділити їх основні функції та структурні одиниці.

4) Проведено аналіз сучасних алгоритмів пошуку інформації, що дозволило виділити їх переваги та недоліки та обрати необхідну цифрову бібліотеку функцій для програмної реалізації модуля;

5) Спроектовано та здійснено моделювання структури додатку пошуку інформації на основі текстових запитів, що дозволило програмно його реалізувати;

6) Програмно реалізовано систему онлайн пошуку інформації на основі текстових запитів, проведено її тестування та порівняльний аналіз з програмами-аналогами, що дозволило проілюструвати високий потенціал розробки.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сергійчук В.В., Жаборинська І.В. Алгоритм машинного перекладу текстових даних на основі онлайн-сервісів веб-браузерів. Збірник тез IV Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», Тернопіль, 02 червня 2021 р. с. 28.

2. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Бакалавр” спеціальності 123 «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ЗУНУ, 2020. 60с.

3. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем»/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.

4. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.

5. Бартлетт Н. Программирование на С++ Путеводитель. The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 2016. 116с.

6. Вебер Дж. Технология С++ в подлиннике. QUE Corporation, 2016, "BNV-Санкт-Петербург", 2017. 256с.

7. Волш А. И. Основы программирования на С++ для World Wide Web. IDG Books Worldwide, Inc., 1996, Издательство "Диалектика", 2016. 458с.

8. Марков А. С. «Базы данных. Введение в теорию и методологию. Финансы и статистика». 2016. С.24-35.

9. Абрамов С. А. Задачи по программированию. М.: Наука, 2018. 256с.

10. Березин Б.И., Начальный курс Delphi. М.: ДИАЛОГ-МИФИ, 2016. 331с.

11. Бондарев В.М. Основы программирования. Харьков: Фолио, Ростов н/Д: Феникс, 2017. 446с.

12. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 2019. 345с.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

- 13.Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие. Пермь: Перм. техн. ун-т, 2014. 516с.
- 14.Грогоно П. Программирование на языке Delphi. М.: Мир, 2012. 216с.
- 15.Дагене В.А. 100 задач по программированию. М.: Просвещение, 2013. 106с.
- 16.Джамса К. Библиотека программиста Java. Jamsa Press, 2016, ООО "Попурри", 2016. 656с.
17. Марков А. С. «Базы данных. Введение в теорию и методологию. Финансы и статистика». 2016. Р. 24-35.
- 18.Заварыкин В.М. Основы информатики и вычислительной техники. М.: Просвещение, 2019. 556с.
- 19.Касаткин В. Н. Информация. Алгоритмы и примеры. ЭВМ. М.: Просвещение, 2011. 219с.
- 20.Кен А. Язык программирования Delphi. Addison-Wesley Longman,U.S.A.,1996, Издательство "Питер-Пресс", 2017. 378с.
- 21.Керниган Б. Язык программирования Delphi. Пер. с англ. М.: Финансы и статистика, 2012. 391с.
- 22.Ляхович В.Ф. Руководство к решению задач по основам информатики и вычислительной техники. М.: Высшая школа, 2014. 127с.
- 23.Мейнджер Дж. Delphi Основы программирования. McGraw-Hill,Inc.,1996, Издательская группа BHV, Киев, 2017. 346с.
- 24.Миков А. И. Информатика. Введение в компьютерные науки. Пермь: Изд-во ПГУ, 2018. 442с.
- 25.Могилев А. В. Информатика: Учеб. пособие для студ. пед. Вузов. М.: Изд. центр «Академия», 2019. 629с.
- 26.Нотон П. JAVA:Справ.руководство. М.:БИНОМ:Восточ.Кн.Компания, 2016: Восточ.Кн.Компания. 447с.
- 27.Нотон П. Полный справочник по Java. McGraw-Hill,1997, Издательство "Диалектика",2017. 556с.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

28.Ренеган Э.Дж. 1001 адрес WEB для программистов :Новейший путеводитель программиста по ресурсам World Wide Web:Пер.. Минск:Попурри, 2017. 512с.ил.

29.Родли Дж. Создание Java-апплетов. The Coriolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.", 2016. 466с.

30.Секреты программирования для Internet на Java. Ventana Press, Ventana Communications Group, U.S.A., 2016, Издательство "Питер Пресс", 2017. 396с.

31.Семакина И. Г. Информатика. Задачник-практикум: В 2 т.. М.: Лаборатория Базовых Знаний, 2019. 476с.

32.Сокольский М.В. Все об Intranet и Internet. М.:Элиот, 2018. 254с.

33.Тассел Д. Стил, разработка, эффективность, отладка и испытание программ. М.: Мир, 2011. 56с.

34. Тюрин Ю.Н. Анализ данных на компьютере. М.: ИНФРА-М, Финансы и статистика, 2015. 384с.

35.Флэнэген Д. Java in a Nutshell. O'Reilly & Associates, Inc., 1997, Издательская группа BHV, Киев, 2018. 473с.

36.Чен М.С. Программирование на C++:1001 совет:Наиболее полное руководство по Java и Visual J++ :Пер.с англ. Минск:Попурри, 2017. 640с.ил.

37.Эферган М. C++: справочник. QUE Corporation, 2017, Издательство "Питер Ком", 1998. 256с.

38.G. Yang. «Human face detection in a complex background. Pattern Recognition », 27 (1): 2014. P.53-63.

39.Kotropoulos C. «Acoustics, Speech, and Signal Processing», 2017. ICASSP-97, 2017. IEEE International Conference on pp.2537-2540 v. 4

40.Leung TK. «Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching» 2015.p P.83-95.

41.Yow KC. Feature-based human face detection. Image and vision computing 15 (9), 2017. P.713-735.

42.Sinha, P. Perceiving and Recognizing threedimensional forms. PhD thesis, Massachusetts Institue of Technology, 2016. 278p.

					КР.КІ.110325/17.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

43.Lanitis, A «Image Anal. Classifying variable objects using a flexible shape model »Image Processing and its Applications, 2015., P.70-74.

44.Viola P. «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2011., vol. 1, 518p.

45.Jones MJ. «Robust real-time face detection», International Journal of Computer Vision, vol. 57, no. 2, 2014., P.137-154.

46.Buchatskiy AN. «Selection of the Optimal Color Space for Reducing False Positives Rate in the Viola-Jones Method», Актуальні проблеми інфотелекомунікацій в науці та освіті, II Міжнародна науково-технічна та науково-методична конференція. Санкт-Петербург, 2013.

47.Ethan R. ORB: an efficient alternative to SIFT or SURF. Computer Vision (ICCV), IEEE International Conference on. IEEE, 2011. P. 2564–2571.

48.Stefan L. BRISK: Binary Robust Invariant Scalable Keypoints. Computer Vision (ICCV), 2011. P. 2548–2555.

49.Pablo F. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In British Machine Vision Conference (BMVC), 2013.

50.Martin A. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. Of the ACM24: 2001. P.381–395,.

51.Патин М.В. Сравнительный анализ методов поиска особых точек и дескрипторов при группировке изображений по схожести содержания. Молодой ученый. 2016. №11. С. 214-221.

52.Гонсалес Р. Цифровая обработка изображений в среде MATLAB //М.: Техносфера. 2016. 616с.