

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
(повна назва)

Кафедра _____ електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський)

Технологія створення кросплатформеної
інформаційної системи для персональної
багатофункціональної електронної бібліотеки
(тема)

Виконав:

студент _____ ІІ _____ курсу, групи _____ СПм-20-1
Скормонний О.Г.
(прізвище, ініціали)

Спеціальність _____
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма _____
Системне програмування
(повна назва освітньої програми)

Керівник: _____ доц. Носик А.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

_____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Скоромному Олегу Григоровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Технологія створення кросплатформеної інформаційної системи для
персональної багатофункціональної електронної бібліотеки _____

затверджена наказом по університету від “ 5 ” листопада 2021 р. № 1657 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 13 грудня 2021 р.

3. Вхідні дані до роботи _____ 1) платформа .NET Core API пошукових систем, технічне

_____ 2) інструмент побудови графічних інтерфейсів WPF _____

_____ 3) СУБД MSSQL _____

4. Перелік питань, що потрібно опрацювати у роботі _____

_____ 1) аналіз предметної області _____

_____ 2) вибір програмного забезпечення _____

_____ 3) аналіз методів пошуку _____

_____ 4) модифікація методу пошуку _____

_____ 5) розробка інформаційної системи для персональної електронної бібліотеки _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація - 12 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз персональних електронних бібліотек	09.11.21-12.11.21	
2	Вибір програмного забезпечення	13.11.21-16.11.21	
3	Аналіз та модифікація методів пошуку	17.11.21-22.11.21	
4	Розробка програмного забезпечення	23.11.21-28.11.21	
5	Оформлення результатів атестаційної роботи	29.11.21-02.12.21	
6	Подання кваліфікаційної роботи керівникові та її попередній захист	03.12.21-04.12.21	
7	Подання кваліфікаційної роботи на рецензування	05.12.21-06.12.21	

Дата видачі завдання 8 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Носик А.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 62 с., 7 рис., 2 табл., 1 дод., 20 джерел.

VISUAL STUDIO, WINDOWS PRESENTATION FOUNDATION, C#, .NET CORE, ЕЛЕКТРОННІ БІБЛІОТЕКИ, ПОШУКОВІ СИСТЕМИ, SPHINX.

Метою кваліфікаційної роботи є аналіз технологій створення багатоплатформової інформаційної системи для персональної багатofункціональної електронної бібліотеки та розробка власної технології створення інформаційної системи для персональної електронної бібліотеки.

У ході виконання кваліфікаційної роботи проведено огляд сучасних пошукових систем та баз даних з вбудованим повнотекстовим пошуком.

За результатами проведених досліджень були проаналізовані технології та інструменти розробки програмного забезпечення та на їх основі було зроблено обґрунтований вибір.

Як результат кваліфікаційної роботи було створено спеціалізований програмний продукт, що задовольняє вимогам. Програмний засіб було реалізовано на мові C# на платформі .NET в середовищі Visual Studio з графічним інтерфейсом, побудованим за допомогою WPF.

ABSTRACT

Master's thesis: 62 pages, 7 figures, 2 table, 1 appendice, 20 sources.

VISUAL STUDIO, WINDOWS PRESENTATION FOUNDATION, C#,
.NET CORE, ELECTRONIC LIBRARIES, SEARCHING SYSTEMS, SPHINX.

The major goal of this thesis is to analyze technologies for creating a multi-plate information system for a personal multifunctional electronic library and developing its own technology for creating an information system for a personal electronic library.

In the course of the attestation work, an overview of modern search engines and databases with built-in full-text search was conducted.

According to the results of research conducted, technologies and tools for developing software and based on their basis were based.

As a result of attestation work, a specialized software product that meets requirements was created. The software has been implemented in the language # on the .NET platform in the Visual Studio environment with a graphical interface built using WPF.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКРОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 АНАЛІЗ СТАНУ ПРОБЛЕМИ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ПЕРСОНАЛЬНОЇ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ.....	10
1.1 Основні поняття	10
1.2 Сучасний стан електронних бібліотек	10
1.3 Правова складова електронних бібліотек.....	15
2 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	20
2.1 Загальна інформація про системи керування бази даних	20
2.2 Розгляд можливих систем керування базами даних	23
2.3 Аналіз інструментів з розробки програмного забезпечення	26
2.3.1 Аналіз програмних платформ і технологій	27
2.3.2 Аналіз сучасних мов програмування	29
2.3.3 Аналіз сучасних інтегрованих середовищ розробки.....	30
2.3.4 Аналіз інструментів створення графічного інтерфейсу.....	31
3 РЕАЛІЗАЦІЯ МЕТОДІВ ПОШУКУ	33
3.1 Загальні відомості	33
3.1.1 Пошуковий індекс.....	35
3.1.2 Синтаксичний аналіз.....	39
3.2 Визначення пошукових систем.....	41
3.3 Реалізація модифікованого методу пошуку	46
4 ІНСТРУКЦІЯ КОРИСТУВАЧА	50
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	55
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних.

ІФЛА (The International Federation of Library Associations and Institutions, IFLA) – міжнародна організація, яка представляє інтереси бібліотечних та інформаційних працівників.

СКБД (Система керування базами даних) – це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

API (application programming interface) – опис способів, якими одна комп'ютерна програма може взаємодіяти з іншою програмою.

JSON (JavaScript Object Notation) – це структурований текстовий формат обміну даними між комп'ютерами.

MVC (Model-View-Controller) – схема поділу програмних систем на три окремих компоненти: модель, уявлення і контролер.

NoSQL (not only SQL) – термін, що позначає ряд підходів, спрямованих на реалізацію систем управління нереляційними базами даних.

SOLID – це аббревіатура складена з перших літер п'яти базових принципів об'єктно-орієнтованого програмування.

SQL (Structured query language) – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних.

WPF (Windows Presentation Foundation) – графічна підсистема в складі платформи .NET.

XML (Extensible Markup Language) – стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками.

ВСТУП

В наш час велика кількість інформації зберігається в документах. Навіть якщо поставити собі ціль прочитати всі документи, що існують в світі, то це неможливо зробити за життя людини. Також пам'ять людини не є ідеальною, тому не всі прочитані документи можна згадати. Тому слід вести списки прочитаних документів, щоб було легше знаходити джерела своїх знань, згадувати певні моменти, не витрачати час на перечитування забутих документів, що не виявились достатньо корисними.

Для покращення цієї ситуації існують бібліотеки, що зберігають записи про документи, що людина у них брала, але ця система не є централізованою. Для складання списку прочитаних документів в цьому випадку, потрібно обійти всі бібліотеки в яких була взята документи. При цьому частина даних може бути вже втрачена.

Наступним кроком розвитку бібліотек стає їх перехід у електронний простір. Але це не зовсім вирішує проблеми контролю за прочитаними документами. Тому слід виокремити інформаційні системи для персональних електронних бібліотек. На даний час є лише невелика кількість програм, що дозволяє створювати свої персональні електронні бібліотеки.

Одною із проблем інформаційних систем для персональних електронних бібліотек є швидкість пошуку потрібних документів по певним словам, так як, окрім прочитання документів, може потребуватися і згадування окремих моментів, що потребує швидкого пошуку.

Метою кваліфікаційної роботи є аналіз технологій створення багатоплатформової інформаційної системи для персональної багатфункціональної електронної бібліотеки та розробка власної технології створення інформаційної системи для персональної електронної бібліотеки з розробленим для її демонстрації програмного забезпечення. Об'єкт дослідження – процес керування персональною електронною бібліотекою.

Предметом дослідження є методи та засоби розробки багатоплатформового програмного забезпечення для реалізації інформаційної системи для персональної багатофункціональної електронної бібліотеки.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз вимог інформаційних систем для персональних електронних бібліотек;
- обрати оптимальне програмне забезпечення;
- провести аналіз методів та алгоритмів повнотекстового пошуку;
- розробити та побудувати модель кросплатформенного програмного забезпечення;
- розробити кросплатформенну інформаційну систему для персональної багатофункціональної електронної бібліотеки.

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ПЕРСОНАЛЬНОЇ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ

1.1 Основні поняття

Інформаційна система – комунікаційна система, що забезпечує збирання, інформаційний пошук, оброблення, передавання та зберігання інформації [1].

Пошукова система – автоматизована система, призначена для шукання та отримання необхідних документів, текстів, інформаційних об'єктів, які за певними ознаками відповідають інформаційним запитам.

Інформаційна система, призначена для накопичення, упорядкування, обліку, обробляння, зберігання, керування та використання електронних документів і для обслуговування користувачів бібліотеки через телекомунікаційні мережі [1].

Інформаційний запит – інформаційна потреба користувача інформації, подана природною мову.

Інформаційний пошук – шукання, відбирання та надання за певними критеріями інформації, яка відповідає заздалегідь заданим вимогам в інформаційному запиті [1].

База даних (БД) – сукупність даних, організованих за певними правилами, що передбачають загальні принципи опису, зберігання і маніпулювання даними, незалежна від прикладних програм [2].

1.2 Сучасний стан електронних бібліотек

На сьогоднішній день загальновизнаним є той факт, що найбільш ефективно інформаційно-бібліотечне обслуговування населення досягається шляхом створення електронних бібліотек. Електронні бібліотеки можуть

складатися з колекцій електронних документів, так і являти собою систему з уніфікованим підходом до виробництва, зберігання, організації різноманітної інформації. Останнім часом з'являється безліч тлумачень терміна «електронна бібліотека». Під електронною бібліотекою розуміється інформаційна система, що дозволяє надійно зберігати і ефективно використовувати різноманітні колекції електронних документів [2].

Електронна бібліотека – це організоване зібрання документів, що дозволяє здійснювати швидкий і ефективний пошук необхідної інформації. Основні завдання електронної бібліотеки – інтеграція інформаційних ресурсів і ефективна навігація в них. Електронна бібліотека виконує наступні функції: забезпечення схоронності друкованого матеріалу з бібліотечних фондів; створення умов для надання локального і віддаленого доступу користувачів до інформаційних ресурсів; розширення інформаційних послуг; вирівнювання рівнів технологічного розвитку різних установ. В електронних бібліотеках повинні бути передбачені можливості введення або видалення інформаційних об'єктів, їх інтеграції, реструктуризації та інші подібні операції [5].

Створення і використання електронних бібліотек реалізується через накопичення, зберігання, облік і структурування електронної інформації; організацію навігації у інформаційному просторі, доступному через дану електронну бібліотеку; забезпечення ефективного доступу до неї користувачів через телекомунікаційні мережі, а також через навчання користувачів [2].

Сьогодні бібліотеки України не можуть похвалитися значними досягненнями та високим рівнем фінансування програм створення електронних бібліотек. Однак ентузіазм тримається на найвищому рівні. З початку 90-х років 24 бібліотеки розробляли різноманітні за тематикою проекти щодо інформатизації, фінансування яких дуже часто було комбінованим і відбувалося, як шляхом залучення бюджетних коштів, так і за рахунок спонсорської допомоги, грантів [3].

Для успішної роботи електронних бібліотек необхідно вирішити наступні проблеми:

Технічні:

- розробка технології формування інформаційного фонду;
- проблеми ідентифікації та опису інформації - раціональний вибір інформаційних об'єктів, створення ефективної системи метаінформації;
- проблеми зберігання і поширення інформації [4].

Організаційні:

- необхідність створення професійних колективів для розробки та експлуатації електронної бібліотеки, що включають в себе фахівців різних профілів;
- нестача кваліфікованих кадрів і практично повна відсутність місць їх підготовки.

Економічні:

- висока вартість розробки електронної бібліотеки;
- висока вартість обладнання;
- трудомісткість процесу формування та експлуатації фонду [4].

Юридичні:

- визначення статусу електронної бібліотеки – що являє собою електронна бібліотека, якими вона володіє правами і обов'язками, хто визнає деяку інформаційну систему в якості електронної бібліотеки;
- визначення статусу інформації в електронній бібліотеці – забезпечення достовірності інформації, її автентичності оригіналу, автентичності, в тому випадку якщо вона існує тільки в електронному вигляді; до цього ж кола питань відноситься і проблема електронного підпису;
- проблеми комерційного використання інформації, що міститься в електронній бібліотеці;
- проблеми конституційного права громадян на отримання інформації та обмеження на поширення певних видів інформації [4].

Соціальні:

- неготовність наукового товариства брати участь у створенні електронних бібліотек і застосовувати їх у своїй роботі;
- малий досвід викладачів і студентів у використанні електронних бібліотек та інших інформаційних систем в рамках освітнього процесу [4].

Для бібліотеки розвиток напрямку електронних бібліотек означає, перш за все, сканування та формування частини своєї колекції в електронному вигляді, її організацію і обслуговування локальних і віддалених користувачів. Електронні бібліотеки покликані істотно розширити спектр послуг, пропонованих користувачам бібліотек [2].

Електронні бібліотеки, також як і традиційні, можуть бути універсальними і спеціалізованими. Однак, якщо в традиційній бібліотеці спеціалізація визначається тільки підбором літератури, то в електронній – ще й додатковим набором програмних засобів, що дозволяють здійснювати ті чи інші маніпуляції з об'єктами зберігання [6].

Технічно електронна бібліотека реалізується на персональних комп'ютерах, з'єднаних в локальну мережу, яка працює під управлінням мережевої операційної системи. Тексти документів, отримані з видавництв, перевіряються оператором і коректорами і порівнюються з друкованими виданнями з метою подальшого їх використання читачами. Перевірені електронні тексти документи архівуються і включаються в електронну бібліотеку. Додатковим джерелом отримання електронних текстів документів, крім видавництв, служить комп'ютерна мережа [2].

Ефективність електронної бібліотеки:

- вирішуються головні бібліотечні проблеми: мала кількість видань, нестача площ для зберігання фонду, збереження книжкового фонду, використання електронних копій запобігає погіршенню стану оригінальних документів;
- повніше задовольняються інформаційні запити користувачів бібліотеки, користувач отримує інформацію незалежно від часу і місця

знаходження;

- істотно підвищується оперативність надання користувачам необхідних документів і даних. Для більшості користувачів електронна форма надає єдину можливість отримати необхідний документ;
- формується новий імідж бібліотеки, що надає інформацію на нетрадиційних носіях; підвищується значення бібліотечної роботи;
- підвищується рівень інформаційної культури і комп'ютерної грамотності як читачів, так і співробітників бібліотеки;
- використання мультимедійних комп'ютерів, що надають текстову, аудіо та відео інформацію дозволяє краще засвоювати матеріал, бо інформація сприймається комплексно, кількома органами чуття одночасно [2].

Як показує аналіз світової практики, використання електронних видань є одним з магістральних напрямків в сучасній діяльності бібліотек, що дозволяє подолати проблеми з підпискою на періодичні видання в традиційному вигляді. Для вирішення цього завдання бібліотеки об'єднуються в консорціуми, які отримують спільний доступ до пакетів журналів певного видавництва або тематики на фінансових умовах, значно більш вигідних, ніж при виписці друкованих видань кожної бібліотекою окремо [2].

Сьогодні до двох основних бібліотечних функцій додається нова - зосередження в бібліотеках електронних інформаційних ниток, які ведуть до будь-яких джерелами. Тепер бібліотеці можна не прагнути отримати джерело інформації, якщо вона точно знає, де і як його знайти, тобто змінюється сутність поняття «фонди». Модель доступу до інформації стає основною перспективною характеристикою електронної бібліотеки. По-перше, тільки доступ до електронних ресурсів через спеціально організовані електронні бібліотеки може забезпечити необхідну повноту інформування. По-друге, тільки проблемно-орієнтовані електронні бібліотеки, в якій ресурси систематизовані на основі детальних ієрархічних класифікацій, можуть

забезпечити оперативний і якісний пошук необхідного ресурсу [2].

Зараз всі електронні бібліотеки створюються на основі:

- цілеспрямованого оцифровування або оцифровування на замовлення;
- комп'ютерної підготовки видань, що готуються для поширення в традиційної та електронної середовищі;
- технологій для функціонування тільки в електронному середовищі.

Все це - технології підготовки електронних ресурсів. І всі ці підходи чітко асоціюються з традиційним терміном «комплектування» і спрямовані на виконання цієї бібліотечної функції [2].

Рівень розвитку інформаційних сервісів і доступу до ресурсів бібліотек є відображенням рівня розвитку інформаційних технологій в суспільстві в поєднанні з законодавчими нормами держави і світу. Впровадження сучасних технологій в бібліотеки створило умови для розвитку інноваційних сервісів для користувачів [2].

1.3 Правова складова електронних бібліотек

В останні роки ресурси, раніше відкриті тільки для читачів, які прийшли в бібліотеку, завдяки мережевим технологіям стали в значній частині доступні для тих, хто знаходиться на відстані від бібліотек або обмежений в пересуванні зважаючи на складну роботу або за станом здоров'я. Існують певні обмеження в рівнях доступу до ресурсів, причому ці обмеження створюється не бібліотека, вона лише розвиває свою діяльність у правовому полі і прагне у всьому бути законослухняною [2].

Багато бібліотек мають в своїх фондах електронні інформаційні ресурси власної генерації, популярної для всіх бібліотек є і агрегація електронних періодичних видань, які практично витісняють «паперові» періодичні видання з книжкових полиць. Як відомо, існує значна кількість журналів, що відносяться до системи відкритого доступу, але тим не менше найбільш потрібними сьогодні є журнали, доступ до яких обмежується

умовами ліцензійних угод, підписаних безпосередньо між бібліотекою і постачальником послуг доступу, власником ресурсу.

Організаційно-технологічні схеми надання доступу окремим бібліотекам, університетам, а також їх комплексів спільно з їх відділеннями або філіями є не менш складною проблемою, адже не завжди очевидно, де проходить риса правових обмежень і як вирішувати виникаючі проблеми [2].

У сучасному світі все частіше звучать слова на захист права людини на отримання інформації, про неприпустимість бар'єрів, що перешкоджають доступу до знань. Юридичні бар'єри гальмують розвиток суспільства знань і позбавляють людей його переваг. Тому бібліотеки в світі об'єднують свої зусилля, для того щоб адаптувати ключові юридичні норми до здійснення своєї місії в цифрову епоху. І якщо з боку захисників права на доступ до інформації стоять бібліотеки, то з іншого боку - автори і правовласники, які вимагають дотримання закону, що обмежує використання інформаційних ресурсів в електронному вигляді.

Бібліотека має в своїх фондах електронні продукти, значна частина яких є ліцензійними ресурсами, наданими користувачам в Відповідно до ліцензійних угод з правовласниками [2].

Існує ще одна сторона проблеми - бібліотеки не завжди розуміють, що в ліцензії зазвичай обумовлений доступ до електронних матеріалів на чітко визначений період часу і його використання можливо лише на умовах ліцензії. В зобов'язання, які має виконувати бібліотека, найбільш часто входять:

- забезпечення доступу до ліцензійних матеріалами через захищену мережу тільки авторизованим користувачам;
- розробка та використання заходів для ознайомлення авторизованих користувачів до вимог ліцензійних договорів;
- в разі виявлення порушення ліцензійних вимог – негайне повідомлення про це власнику ресурсу, розслідування обставин порушення, вжиття заходів для того, щоб порушення не повторювалося [2].

Обсяги вивантажених інформації контролюються власниками ресурсів з використанням спеціальних програмних засобів. Так, при роботі з електронними журналами здійснюється контроль за тим, які статті вивантажуються. Як тільки надходить сигнал про перевищення допустимих обсягів скачування, доступ до ресурсу припиняється [2].

Одна з основних проблем пов'язана з користувачами ресурсів, що не розуміють, що таке вимоги ліцензійного договору, обмеження в умовах копіювання документів. Незважаючи на всі вжиті заходи читач починає копіювати себе статті всього номера журналу, а то й кількох номерів.

Реакція організації, що надала доступ, буває дуже швидкою. Порухення ліцензійної угоди призводить до відключення бібліотеки від даного ресурсу.

Відновлення доступу можливо тільки після переговорів, виявлення порушника, проведення профілактичної роботи або каральних заходів, звіту про проведену роботу [2].

ІФЛА - міжнародна недержавна організація, яка існує для того, щоб робити, підтримувати і координувати наукові дослідження, вивчення і поширення інформації щодо всіх аспектів бібліотечної та інформаційної роботи у всьому світі, а також організовувати дискусії і навчання в цій області. В міжнародному обговоренні питань забезпечення авторського права ІФЛА представляє інтереси бібліотек світу і їх користувачів. Законодавство в галузі забезпечення авторського права поширюється на багато аспектів діяльності бібліотек [7].

Воно визначає види послуг, що надаються бібліотеками своїм користувачам, і умови, при дотриманні яких бібліотеки можуть надати доступ до матеріалів, що охороняються авторським правом. Воно впливає на дії бібліотек, що виконують роль навігаційних посередників і здійснюючих заходи для ефективного архівування та зберігання матеріалів. Саме з цих причин ІФЛА бере участь в міжнародних дебатах щодо забезпечення авторського права [7].

ІФЛА підтримує баланс між інтересами сторін, який сприяє прогресу суспільства в цілому, забезпечуючи надійну, ефективну захист інтересів власників прав і прийнятних умов доступу для того, щоб заохочувати творчість, інновації, дослідження, освіту і навчання. Вона підтримує ефективну реалізацію авторського права і визнає, що бібліотеки відіграють важливу роль як в контролі за доступом, так і в полегшенні доступу до зростаючої кількості місцевих та віддалених електронних інформаційних ресурсів. Бібліотекарі та інформаційні працівники сприяють шанобливого ставлення до авторського права і активно захищають роботи, які охороняються авторським правом, від піратства і неправомірного, несанкціонованого використання їх як в друкованому, так і цифровому просторі [7].

Загальновизнано, що протягом тривалого періоду часу бібліотеки виконують роботу, пов'язану з інформуванням користувачів про важливість забезпечення авторського права і заохочують їх дії на виконання його. Разом з тим, ІФЛА дотримується думки, що надмірне регулювання авторського права може загрожувати демократичним традиціям і зробити негативний вплив на дотримання принципів соціальної справедливості, необгрунтовано обмежуючи доступ до інформації та знань [7].

Все більше і більше інформації проводиться в цифровому форматі. Нові комунікаційні технології несуть з собою безпрецедентні можливості для поліпшення доступу до інформації та мають достатній потенціал для забезпечення зв'язку і доступу для тих, хто має обмеження, які визначаються відстанню або економічними обставинами. Однак, сьогодні ми знаємо, що ця технологія має потенціал також для того, щоб ще більше розділити суспільство на тих, хто володіє інформацією і тих, хто не володіє інформацією. Якщо прийнятний доступ до робіт, захищених авторським правом, нічого очікувати підтримуватися в цифровому середовищі, то в подальшому це призведе до появи бар'єру перед тими, хто не зможе дозволити собі платити за доступ.

В інформаційному суспільстві бібліотеки продовжать відігравати вирішальну роль в забезпеченні всіх доступом до інформації. Належним чином функціонують національні та міжнародні мережі бібліотек та інформаційних служб є головним компонентом у забезпеченні доступу до інформації. Традиційно, бібліотеки в стані забезпечити доступ на прийнятних умовах до придбаних і зберігаються в їх колекціях екземплярам робіт, захищених авторським правом. Однак, якщо в майбутньому весь доступ і використання інформації в цифровому форматі стануть платними, можливості бібліотек щодо забезпечення доступу їх користувачів до інформації будуть сильно обмежені. Для того, щоб зберегти рівновагу між інтересами власників прав і користувачів, ІФЛА розробила наступну заяву про принципи [7].

У 1996 р члени Всесвітньої організації інтелектуальної власності прийняли два міжнародні договори з метою оновити законодавство про авторське право для випадку цифрового простору. В підтвердження того, що вже існуючі винятки та обмеження можуть бути перенесені в цифрову середу, країни, що входять в WIPO, відхилили вимогу про те, що "Цифрове є іншим". Сторонам, яка укладає договір, дозволяється розвивати і розширювати такі обмеження в цифровому середовищі, а також створювати нові винятки при необхідності [7].

ІФЛА дотримується позиції, що якщо бібліотекам і громадянам не гарантовані винятки, що дозволяють мати доступ і використовувати матеріали без оплати для цілей, що становлять суспільний інтерес, і відповідно до справедливої практики, такий як освіта і наукові дослідження, існує небезпека, що тільки ті, хто може дозволити собі платити, будуть мати перевагу і зможуть скористатися благами Інформаційного суспільства. Це як ніколи раніше призведе до поділу суспільства на інформаційно багатих і інформаційно бідних. Далі, не повинно бути ніякої дискримінації в законодавстві про авторське право щодо людей, що мають фізичні обмеження [7].

2 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Загальна інформація про системи керування бази даних

База даних – набір даних, організованих певними правилами, що забезпечують загальні принципи опису, зберігання та маніпулювання даних.

Основні компоненти СУБД (рисунок 2.1):

- інструменти представлення даних у базі даних;
- інструменти маніпулювання даних;
- користувацькі інтерфейси, адміністратор баз даних та комунікацій.

Інструменти презентації даних у поєднанні з інтерфейсами організовуються базою даних на різних рівнях абстракції даних. Існує три рівні абстракції та, відповідно, представлення даних:

- фізичний;
- концептуальний;
- зовнішній [8].

Інструменти фізичного рівня враховують характеристики конкретної публікації, форматів та способів запису даних про фізичні середовища. Засоби концептуального рівня відображають дані фізичного рівня з певною абстракцією.

Концептуальний рівень визначає модель презентації даних у базі даних. В даний час застосовуються три моделі представлення даних у базі даних:

- ієрархічна;
- мережа;
- реляційна.

Інструменти зовнішнього рівня перетворюють дані у форму, зрозумілий для користувача, і з яким користувач може працювати через його інтерфейси.

Залежно від призначення СУБД та моделі презентації даних у базі даних, користувачу може бути надано такі інтерфейси:

- меню;
- командний;
- багатовіконний;
- графічний;
- стандартні мови запиту та маніпуляції даних;
- генератор звітів.

Додаткові інтерфейси пропонуються адміністратору бази даних для виконання своїх функцій:

- мови програмування високого рівня;
- мови СУБД;
- генератори інтерфейсу користувачів.

Інтерфейси зв'язку забезпечують експорт та імпорт даних в інші програми та комп'ютерні мережі [8].

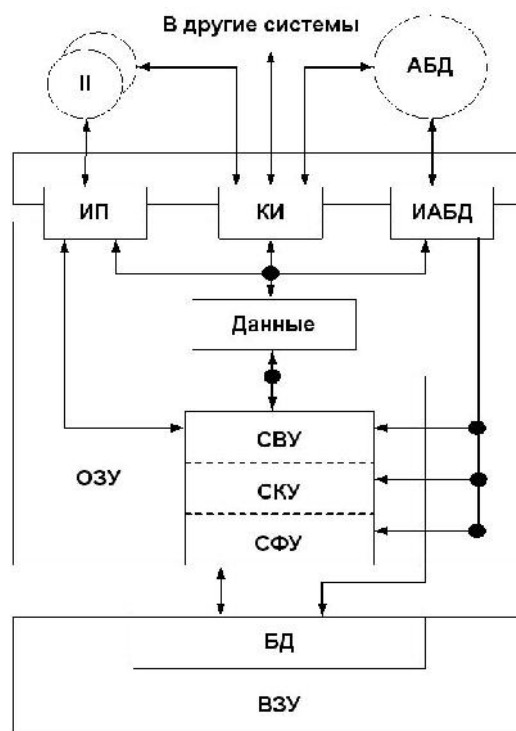


Рисунок 2.1 – Архітектура системи керування базою даних

Презентація даних у базі даних на різних рівнях забезпечується програмами різних рівнів машинної залежності. Зовнішній рівень використовує мови запитів та маніпуляції даних та мови програмування високого рівня. Машинно-залежні мови необхідні для роботи на фізичному рівні. Концептуальний рівень вимагає використання обох мов.

Інтерфейси СУБД за своїм призначенням та рівнем поділяються на інтерфейси призначені для користувача, інтерфейси адміністратора баз даних і інтерфейси зв'язку. Реалізація конкретних інтерфейсів в СУБД визначається призначенням системи, моделлю представлення даних в БД, орієнтацією готовності користувача та іншими факторами [8].

Мова запитів і маніпулюваннями даними в БД є основним інтерфейсом для створення і підтримки бази даних. Найбільш доопрацьованою і добре відомою є мова SQL, розроблена IBM для роботи з реляційними базами даних. Мова SQL приймається як стандарт при розробці СУБД американським національним інститутом стандартів і міжнародною організацією зі стандартизації.

Мова бази даних є частиною СУБД, за допомогою якого створюються прикладні програми для роботи з інформацією в БД.

Комунікаційні інтерфейси забезпечують експорт та імпорт даних в інші системи застосування комп'ютерних мереж.

Залежно від організації структури баз даних і інтерфейсів, що надаються користувачам, СУБД ділиться на наступні групи.

Однофайлові СУБД – найпростішим в зверненні системи, які не вимагають великих ресурсів комп'ютера. Зазвичай застосовуються для телефонних та адресних довідників, списків товарів в магазинах, на складах, організації картотек в бібліотеках [8].

Багатофайлові непрограмовані СУБД забезпечують користувачам широкі можливості для створення бази даних, так як вони дозволяють зберігати і використовувати записи різних типів.

Многофайловий програмовані СУБД надають користувачам

можливості по написанню прикладних програм. Ця СУБД вимагає значних ресурсів комп'ютера і в певний час, щоб отримати навички роботи з базами даних.

Розрахована на багато користувачів СУБД використовується для роботи в локальних комп'ютерних мережах [8].

Широке застосування баз даних і СУБД висунуло ряд вимог, виконання яких найближчим часом повинно істотно спростити взаємодію користувача з базою даних:

- використання для організації баз даних комп'ютерів, що високу продуктивність і великий обсяг пам'яті;
- застосування периферійних пристроїв, що спрощують роботу користувачів з базою;
- збільшення частки багатофайлових і програмованих СУБД, що підтримують реляційні бази даних;
- розвиток стандартних мов запитів та маніпулювання даними;
- удосконалення автоматизованих систем програмування, що працюють в середовищі баз даних під керуванням СУБД;
- створення уніфікованих інтерфейсів для доступу користувачів до баз даних з різних систем програмування, текстових редакторів, електронних таблиць.

2.2 Розгляд можливих систем керування базами даних

Microsoft SQL Server – система керування базами даних (СКБД), розроблена корпорацією Microsoft. Основна використовувана мова запитів – Transact-SQL, створений спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO по структурованого мови запитів (SQL) з розширеннями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СУБД в цьому сегменті ринку [9].

SQL Server в основному побудований навколо структури таблиці на основі рядків, що з'єднує пов'язані елементи даних у різних таблицях один до одного, уникаючи необхідності надмірно зберігати дані в декількох місцях в межах бази даних. Реляційна модель також забезпечує референтну цілісність та інші обмеження цілісності для підтримки точності даних. Ці перевірки є частиною більш широкого дотримання принципів атомарності, консистентності, ізоляції та довговічності, колективно відомі як ACID властивості, і призначені для забезпечення того, що транзакції бази даних обробляються надійно.

Компонент ядра Microsoft SQL Server є система бази даних SQL Server, який керує зберіганням даних, обробку та безпеку. Вона включає в себе реляційну систему, який обробляє команди та запити та системний движок, який керує файлами бази даних, таблицями, сторінками, індексами, буферами даних та транзакцій. Забезпечені процедури, тригери, перегляди та інші об'єкти бази даних також створюються та виконуються системою бази даних [9].

Сидячи під системою бази даних – операційна система SQL Server, або SQLOS. SQLOS обробляє функції нижчого рівня, наприклад, керування пам'яттю та введення / виводу, планування роботи та блокування даних, щоб уникнути конфліктуючих оновлень. Шар мережевого інтерфейсу розташований над системою бази даних та використовує протокол потоку даних Microsoft для полегшення взаємодії запиту та відповіді з серверами баз даних. І на рівні користувача, розробники пишуть T-SQL запити для створення та модифікації структур бази даних, маніпулювати даними, здійснювати захист безпеки та резервні копії баз даних.

PostgreSQL – вільна об'єктно-реляційна система керування базами даних (СКБД) [10].

Переваги PostgreSQL:

- підтримка бази даних необмеженого розміру;
- потужні і надійні механізми транзакцій і реплікації;

- розширювана система вбудованих мов програмування, а також підтримка завантаження C-сумісних модулів;

- спадкування;
- простота розширення.

Поточні обмеження PostgreSQL:

- максимальний розмір таблиці – 32 ТБ;
- максимальний розмір запису – 1,6 ТБ;
- максимальний розмір поля – 1 Гб;
- максимальна кількість полів в записі 250-1600;

Функції в PostgreSQL являються блоками коду, що виконуються на сервері, а не на клієнті БД. Незважаючи на те, що вони можуть бути написані на чистому SQL, реалізація додаткової логіки, така як умовні переходи і цикли, виходить за рамки SQL і вимагають використання деяких розширень мови. Функції можуть бути написані з використанням різних мов програмування. PostgreSQL дозволяє використовувати функції, які повертають набір записів, які потім можуть бути використані таким же чином, як результат звичайного запиту.

Механізм правил в PostgreSQL – механізм створення призначених для користувача обробників не тільки операції маніпулювання даними, а й операцій вибірки. Основна відмінність від механізму тригерів полягає в тому, що правила спрацьовують на етапі розбиття запиту, перш ніж вибрати оптимальний план виконання, а сам процес виконання.

Спадкування в PostgreSQL реалізовано на рівні таблиці. Таблиці можуть успадковувати характеристики і набори полів з інших. У цьому випадку дані додані в згенерованій таблиці будуть приймати участь в запитах до батьківської таблиці.

MongoDB – документоорієнтована NoSQL система керування базами даних [11].

Переваги MongoDB:

- відсутність схеми даних;

- база даних базується на збірках різних документів, тому кількість полів, вміст та розмір цих документів може відрізнятися;
- надзвичайно чітка структура кожного об'єкта;
- гарна масштабованість;
- для збереження даних, що потрібні в роботі, використовується внутрішня пам'ять, яка дозволяє отримувати більш швидкий доступ;
- дані зберігаються як документи JSON;
- підтримує запити динамічних документів;
- відсутність складних запитів;
- відсутня необхідність конвертації об'єктів програми до об'єктів БД.

Для програмного додатку було обрано базу даних Microsoft SQL Server, через її простоту інтеграції з іншими обраними технологіями, потужні можливості та гарну підтримку.

LiteDB – це нереляційна база. Як у SQLite, база представлена одним файлом. Повністю написано на C#. Перевагою являється простота використання: просто потрібно дати об'єкт бібліотеці, а серіалізацію вона бере на себе [12].

Для розроблення програмного засобу було обрано СУБД Microsoft SQL Server.

2.3 Аналіз інструментів з розробки програмного забезпечення

Для виконання роботи потрібно обрати технології, що будуть використані в процесі розробки, а саме:

- програмна платформа;
- мова програмування;
- інтегроване середовище розробки;
- інструменти створення графічного інтерфейсу.

2.3.1 Аналіз програмних платформ і технологій

Активний розвиток ядра .NET став закономірним результатом роботи над програмами. Однією з основних причин головного болю майже кожного розробника є непередбачувані умови, в яких буде запущена програма. Кожен пристрій використовує свою власну версію операційної системи, конфігурації, драйверів тощо, що не завжди дозволяє все заздалегідь розглянути і зробити програму програми [13].

Саме з цієї причини почався пошук способу об'єднання навколишнього середовища для запуску додатків. І одна з найуспішніших спроб у цій галузі це .Net, що була розроблена більш ніж 20 років тому Microsoft.

Основна ідея її створення полягала в тому, щоб додати спеціальну проміжну програму, що буде розташована між операційною системою та програмою. На цю програму були покладені функції згладжування конкретної конфігурації ОС, тим самим спрощуючи програму запуску у існуючому середовищі [13].

.NET активно використовується для розробки додатків наступних типів:

- веб-API та мікрослужби;
- класичні та мобільні програми;
- веб-додатки;
- служби операційної системи;
- консольні програми.

Перевага використання платформи заключається у тому, що файли коду та проекту виглядають ідентично і не залежать від типу розробленого застосування.

Окрім операційної системи Windows, платформа .Net Core також доступна на різних дистрибутивах Linux. Розробники публікують новий випуск практично для всіх дистрибутивів Linux, майже завжди є менеджер пакетів для встановлення. З цієї причини платформа стає все більш

популярною у програмістів та поступово витісняє аналогічні рішення [13].

Платформа Java – це набір програм, що полегшують розробку та запуск програм, написаних на мові програмування Java. Платформа Java включає механізм виконання (називається віртуальною машиною), компілятор та набір бібліотек. Також можуть бути додаткові сервери та альтернативні бібліотеки, які залежать від вимог. Java не відноситься до будь-якого процесора або операційної системи, оскільки платформи Java були реалізовані для широкого спектру апаратних та операційних систем, щоб дозволити програмам Java працювати однаково на всіх [14].

У більшості сучасних операційних систем для спрощення роботи програміста надається великий код повторного використання. Цей код, як правило, надається як набір динамічно завантажених бібліотек, що додатки можуть викликати під час виконання. Оскільки платформа Java не залежить від будь-якої конкретної операційної системи, програми не можуть покладатися на будь-які раніше існуючі бібліотеки ОС. Замість цього платформа Java забезпечує повний набір власних стандартних бібліотек класів, що містять функції багаторазового використання, які зазвичай знаходяться в сучасних операційних системах. Більша частина системної бібліотеки також написана на Java.

Бібліотеки класів Java виконують три завдання на платформі Java. По-перше, як інші стандартні бібліотеки коду, бібліотека Java забезпечують програміста набором відомих функцій для виконання загальних завдань, таких як ведення списків елементів або виконання складного синтаксичного аналізу рядків. По-друге, бібліотеки класів надають абстрактний інтерфейс для завдань, що в значній мірі залежать від апаратного забезпечення та операційної системи. Такі завдання, як доступ до мережі та доступ до файлів, часто переплітаються з різними реалізаціями кожної платформи.

Для розроблення програмного забезпечення було обрано платформу .Net Core.

2.3.2 Аналіз сучасних мов програмування

C# – це об'єктно-орієнтована, проста і одночасно потужна мова програмування, яка дозволяє розробникам створювати багатofункціональні програми [15].

C# відносяться до мов компільованого типу, тому він має всі переваги таких мов. Через широкий спектр синтаксичних конструкцій та вміння працювати з платформою .NET, C # дозволяє швидше, ніж будь-яка інша мова, розробляти програмні рішення.

Велика надійність була досягнута завдяки роботі CLR машини, оскільки, на відміну від інших компіляторів CLR, він запускає розроблену заявку на віртуальний процесор. Тому у випадках будь-яких помилок це не вплине на роботу інших програм у системі, але вона також позначає, що для запуску програми потрібно додатковий час. Відповідно, програма, написана в мові програмування C# більш надійна, але менш швидка.

Visual Basic .NET (VB.NET) – багатopарадигмова, об'єктно-орієнтована мова програмування, реалізована на платформі .NET. Вона має такі переваги:

- об'єктно-орієнтована. Від примітивних типів даних до визначених користувачем типів даних засновано на об'єктах. Він легко дає всі переваги будь-якої мови на базі OOP;
- легкий у навчанні. Через простіший синтаксис та меншу криву навчання легше навчатись;
- високоструктурована мова;
- компонентно-орієнтований. Дотримується концепції компонента тобто, вся програма поділяється на більш прості компоненти, а потім ці компоненти розробляються індивідуально, щоб зробити повну програму [15].

Для програмного засобу було обрано мову C# тому, що вона надійна та дозволяє швидко розробляти додатки.

2.3.3 Аналіз сучасних інтегрованих середовищ розробки

Microsoft Visual Studio – це преміум IDE, вартість якого залежить від редакції і типу підписки. Це дозволяє створювати різні проекти, починаючи від мобільних і веб-додатків і закінчуючи відеоіграми. Microsoft Visual Studio включає в себе безліч інструментів для тестування сумісності – ви можете перевірити додаток на більш ніж 300 пристроїв і браузерів. Завдяки своїй гнучкості, це IDE ідеально підходить як для студентів, так і для професіоналів [16].

Особливостями Visual Studio являються:

- велика колекція розширень, що постійно оновлюються;
- технологія автодоповнення IntelliSense;
- можливість налаштування робочого простору;
- підтримка розділеного екрану.

З недоліків можна виділити важкість цієї IDE. Для того, щоб виконати невеликі зміни, можуть знадобитися значні ресурси, так що якщо вам потрібно виконати просту і швидку задачу, зручніше використовувати більш простий редактор [16].

Eclipse – Це безкоштовне середовище розробки, яка добре підходить як для початківців, так і досвідчених розробників. Окрім інструментів налагодження та підтримки GIT та CVS, Eclipse поставляється з Java та інструментом для створення плагінів. Спочатку Eclipse використовувався тільки для Java, але зараз завдяки плагінам та розширенням його функції значно розширилися. Функціональність Eclipse не настільки велика, як IntelliJ IDEA, але це середовище розробки розповсюджується з відкритим кодом.

Перевагами Eclipse являються:

- можливість програмувати на декількох мовах;
- значна гнучкість середовища через модульність;
- можливість інтеграції JUnit.

До недоліків можна віднести те, що новачкам може бути важко

розібратися в різноманітності можливостей.

Як основний інструментальний програмний засіб було обрано середовище розробки Microsoft Visual Studio тому, що воно зручне, багатофункціональне та має гарну сумісність з обраними вище інструментами.

2.3.4 Аналіз інструментів створення графічного інтерфейсу

Windows Presentation Foundation – це фреймворк для інтерфейсу користувача, що створює програми для настільних клієнтів. Платформа розробки WPF підтримує широкий набір функцій для розробки додатків, включаючи модель програми, ресурси, елементи управління, графіку, макет, зв'язування даних, документи та безпеку.

WPF є частиною .NET. WPF використовує розширену мову розмітки програми, щоб забезпечити описову модель для програмування програми.

Перевагами WPF являються:

- підтримує гнучкий потік, розміщення елементів керування на основі їх вмісту замість фіксації елементів керування в одному постійному місці з деякими постійними конкретними координатами;
- підтримує аудіо та відео;
- не потребує таймерів для оновлення вмісту. Замість цього весь WPF підтримує анімацію, яка працює автоматично;
- робота ведеться за допомогою базових фігур, текстових блоків та інших контролерів, що спрощує завдання рисування інтерфейсу;
- дозволено редагувати інтерфейс безпосередньо з XAML-коду;
- можливе використання стилів та шаблонів.

Windows Forms – технологією інтелектуальних клієнтів для платформи .NET. Вона являється набором керованих бібліотек, що спрощують виконання стандартних завдань, таких як читання з файлової системи та запису до нього. За допомогою середовища розробки ви можете створювати

інтелектуальні клієнтські застосунки Windows Forms, які відображають інформацію, запитують дані від користувачів та обмінюються даними з віддаленими комп'ютерами через мережу.

У Windows Forms, форма являє собою візуальну поверхню, на яку відображається інформація. Зазвичай програма Windows Forms побудована шляхом розміщення елементів керування та запису коду, щоб відповісти на дії користувача, такі як кліки миші або натискання клавіш. Елемент управління - це окремий елемент інтерфейсу користувача, призначений для відображення або введення даних.

При виконанні користувачем будь-якої дії з формою або одним з його елементів керування, створюється подія. Програма відповідає на ці події за допомогою коду та опрацьовує подій, коли вони виникають.

Перевагами WinForms являються:

- велика надійність;
- велика колекція створених компонентів.

Для програмного засобу був обраний WPF тому, що він гнучкий, що дозволяє створювати різноманітні варіанти віконного інтерфейсу.

3 РЕАЛІЗАЦІЯ МЕТОДІВ ПОШУКУ

3.1 Загальні відомості

Пошукова система – алгоритми і реалізуюча їх сукупність комп'ютерних програм, що надає користувачеві можливість швидкого доступу до необхідної йому інформації за допомогою пошуку в великій колекції доступних даних. Одне з найбільш відомих застосувань пошукових систем - веб-сервіси для пошуку текстової або графічної інформації у Всесвітній павутині. Існують також системи, здатні шукати файли на FTP-серверах, товари в інтернет-магазинах, інформацію в групах новин Usenet, документи [17].

Для пошуку інформації за допомогою пошукової системи користувач формулює запит. Робота пошукової системи полягає в тому, щоб за запитом користувача знайти документи, що містять або зазначені ключові слова, або слова, будь-яким чином пов'язані з ключовими словами. При цьому пошукова система генерує сторінку результатів пошуку. Така пошукова видача може містити різні типи результатів, наприклад: веб-сторінки, зображення, аудіофайли.

Деякі пошукові системи також витягають інформацію з відповідних баз даних і каталогів ресурсів в Інтернеті. Для пошуку потрібних відомостей найзручніше скористатися сучасними пошуковими машинами, які дозволяють швидко виявити необхідні відомості і забезпечують точність і повноту пошуку. При роботі з цими машинами досить задати ключові слова, найбільш точно відображають потрібну інформацію, або скласти більш складний запит з ключових слів для уточнення області пошуку. Після введення запиту на пошук ви отримаєте список посилань на документи в Інтернеті, зазвичай звані web-сторінками або просто сторінками, в яких містяться зазначені ключові слова. Зазвичай посилання доповнюються

фрагментами тексту з виявленого документа, які часто допомагають відразу визначити тематику знайденої сторінки [17].

Пошукова система тим краще, чим більше документів, релевантних запиту користувача, вона буде повертати. Результати пошуку можуть ставати менш релевантними через особливості алгоритмів або внаслідок людського фактора [17].

За методами пошуку і обслуговування поділяють чотири типи пошукових систем: системи, що використовують пошукові роботи, системи, керовані людиною, гібридні системи і мета-системи. В архітектуру пошукової системи зазвичай входять:

- пошуковий робот, який збирає інформацію з сайтів мережі Інтернет або з інших документів;
- індексатор, що забезпечує швидкий пошук по накопиченій інформації;
- графічний інтерфейс для роботи користувача [17].

Існує чотири типи пошукових систем: з пошуковими роботами, керовані людиною, гібридні і мета-системи.

Системи, що використовують пошукові роботи. Складаються з трьох частин: краулер, індекс і програмне забезпечення пошукової системи. Краулер потрібен для обходу мережі і створення списків веб-сторінок. Індекс - великий архів копій веб-сторінок. Мета програмного забезпечення - оцінювати результати пошуку. Завдяки тому, що пошуковий робот в цьому механізмі постійно досліджує мережу, інформація більшою мірою актуальна. Більшість сучасних пошукових систем є системами даного типу [17].

Системи, керовані людиною. Ці пошукові системи одержують списки веб-сторінок. Каталог містить адресу, заголовок і короткий опис сайту. Каталог ресурсів шукає результати тільки з описів сторінки, представлених йому веб-майстрами. Гідність каталогів в тому, що всі ресурси перевіряються вручну, отже, і якість контенту буде краще в порівнянні з результатами, отриманими системою першого типу автоматично. Але існує і недолік -

оновлення даних каталогів виконується вручну і може істотно відставати від реального стану справ. Ранжування сторінок не може миттєво змінюватися. Як приклади таких систем можна привести каталог Yahoo, dmoz і Galaxy.

Гібридні системи. Такі пошукові системи, як Yahoo, Google, MSN, поєднують в собі функції систем, що використовують пошукові роботи, і систем, керованих людиною [17].

Мета-системи. Метапоіскова системи об'єднують і ранжирують результати відразу декількох пошукових систем. Ці пошукові системи були корисні, коли у кожної пошукової системи був унікальний індекс, і пошукові системи були менш «розумними». Оскільки зараз пошук набагато покращився, потреба в них зменшилася. Приклади: MetaCrawler і MSN Search.

3.1.1 Пошуковий індекс

Пошуковий індекс – структура даних, яка містить інформацію про документи і використовується в пошукових системах. Індекссування, що здійснюється пошуковою машиною, - процес збору, сортування та зберігання даних з метою забезпечити швидкий і точний пошук інформації. Створення індексу включає міждисциплінарні поняття з лінгвістики, когнітивної психології, математики, інформатики та фізики [18].

Популярні пошукові машини зосереджуються на повнотекстової індексації документів, написаних на природних мовах. Мультимедійні документи, такі як відео і аудіо і графіка, також можуть брати участь в пошуку.

Мета-пошукові машини використовують індекси інших пошукових сервісів і не зберігають локальний індекс, в той час як пошукові машини, засновані на кеш сторінках, довго зберігають як індекс, так і текстові корпуси. На відміну від повнотекстових індексів, частково-текстові сервіси обмежують глибину індексації, щоб зменшити розмір індексу [18].

Мета використання індексу – підвищення швидкості пошуку релевантних документів по пошуковому запиту. Без індексу пошукова машина повинна була б сканувати кожен документ в корпусі, що вимагало б великої кількості часу і обчислювальної потужності.

Додаткова пам'ять, що виділяється для зберігання індексу, і збільшення часу, потрібного для оновлення індексу, компенсується зменшенням часу на пошук інформації [18].

При розробці пошукової системи необхідно враховувати наступні фактори:

- фактори злиття;
- методи зберігання;
- розмір індексу;
- швидкість пошуку;
- зберігання;
- відмовостійкість.

Архітектура пошукової системи різниться за способами індексування і за методами зберігання індексів, задовольняючи фактори. Індокси бувають наступних типів:

- суфіксне дерево. Образно структуроване як дерево, підтримує лінійний час пошуку. Побудовано на зберіганні суфіксів слів. Дереву підтримують розширене хешування, яке важливо для індексації пошукової системи. Використовується для пошуку за шаблоном в послідовності ДНК і кластеризації. Основним недоліком являється те, що зберігання слова в дереві може зажадати простір за межами необхідного для зберігання самого слова. Альтернативне уявлення – суфіксний масив. Вважається, що він вимагає менше віртуальної пам'яті і підтримує стиснення даних [18];

- інвертований індекс. Сховище списку входжень кожного критерію пошуку, зазвичай у формі хеш-таблиць або бінарного дерева;

- індекс цитування. Сховище цитат або гіперпосилань між документами для підтримки аналізу цитування;

- n -грами. Сховище послідовностей довжин даних для підтримки інших типів пошуку або аналізу тексту;
- матриця термів документа. Використовується в латентно-семантичному аналізі, зберігає входження слів в документах в двовимірній розрідженій матриці.

Однією з основних завдань при проектуванні пошукових систем являється управління послідовними обчислювальними процесами. Існує ситуація, в яких можливе створення стану гонки і когерентних відмов. Наприклад, новий документ доданий до корпусу, і індекс повинен бути оновлений, але в той же час індекс повинен продовжувати відповідати на пошукові запити. Це колізія між двома конкуруючими завданнями. Індикатор являється виробником доступною для пошуку інформації, а користувачі, які її шукають, – споживачами. Проблема посилюється при розподіленому зберіганні і розподіленій обробці. Щоб масштабувати великі обсяги індексованої інформації, пошукова система може ґрунтуватися на архітектурі розподілених обчислень, при цьому пошукова система складається з декількох машин, що працюють узгоджено. Це збільшує ймовірність нелогічності і робить складніше підтримку повністю синхронізованої, розподіленої, паралельної архітектури.

Необхідність розробки прямого індексу пояснюється тим, що краще відразу зберігати слова за документами, оскільки їх в подальшому аналізують для створення пошукового індексу. Формування прямого індексу включає асинхронну системну обробку, яка частково обходить вузьке місце поновлення інвертованого індексу. Прямий індекс сортують, щоб перетворити в інвертований. Прямий індекс по суті являє собою список пар, що складаються з документів і слів, відсортоване за документами. Перетворення прямого індексу до інвертованого являється питанням сортування пар по словам. В цьому відношенні інвертований індекс – відсортований за словами прямий індекс [18].

Багато пошукових систем використовують інвертований індекс при

оцінці пошукового запиту, щоб швидко визначити місце розташування документів, що містять слова із запиту, а потім рангувати ці документи за релевантністю. Оскільки інвертований індекс зберігає список документів, що містять кожне слово, пошукова система може використовувати прямий доступ, щоб знайти і отримати документи [18].

Інвертований індекс може тільки визначити, чи існує слово в межах конкретного документа, так як не зберігає ніякої інформації щодо частоти і позиції слова, і тому його вважають логічним індексом. Інвертований індекс визначає, які документи відповідають запиту, але не оцінює відповідні документи.

У деяких випадках індекс включає додаткову інформацію, таку як частота кожного слова в кожному документі або позиція слова в документі. Інформація про позицію слова дозволяє пошуковому алгоритму ідентифікувати близькість слова, щоб підтримувати пошук фраз. Частота може використовуватися, щоб допомогти в рангуванні документів за запитом. Такі теми в центрі уваги досліджень інформаційного пошуку.

Інвертований індекс представлений розрідженою матрицею, так як не всі слова присутні в кожному документі. У деяких випадках індекс представлений у формі двійкового дерева, яка вимагає додаткової пам'яті, але може зменшити час пошуку. У великих індексах архітектура, як правило, представлена розподіленої хеш-таблицею.

Інвертований індекс заповнюється шляхом злиття або відновлення. Архітектура може бути спроектована так, щоб підтримувати інкрементного індексацію, де злиття визначає документ або документи, які будуть додані або оновлені, а потім аналізує кожен документ в слова. Для технічної точності, злиття об'єднує недавно індексовані документи, які зазвичай перебувають у віртуальній пам'яті, з індексним кешем, який знаходиться на одному або декількох жорстких дисках комп'ютера [18].

Після розбору індексатор додає вказаний документ до списку документів для відповідних слів. У більшій пошуковій системі процес

знаходження кожного слова для інвертованого індексу може бути дуже трудомістким, тому його, як правило, поділяють на дві частини:

- розробка прямого індексу;
- сортування прямого індексу в інвертований індекс.

Створення та підтримка великомасштабного пошукового індексу вимагає значної пам'яті і виконання завдань обробки. Багато пошукових систем використовують ту чи іншу форму стиснення, щоб зменшити розмір індексів на диску.

3.1.2 Синтаксичний аналіз

Синтаксичний аналіз документа передбачає розбір документа на компоненти для вставки в прямий і інвертований індекси. Знайдені слова називають токенами, і в контексті індексації пошукових систем і обробки природної мови парсинг часто називають токенізацією. Синтаксичний аналіз іноді називають морфологічним аналізом, контент-аналізом, текстовим аналізом, аналізом тексту, генерацією узгодження, сегментацією мови, лексичним аналізом. Терміни «індексація», «парсинг» і «токенізація» взаємозамінні в корпоративному сленгу [19];

Обробка природної мови постійно досліджується і поліпшується. Токенізація має проблеми з отриманням необхідної інформації з документів для індексації, щоб підтримувати якісний пошук. Токенізація для індексації включає в себе кілька технологій, реалізація яких може бути комерційною таємницею [19].

Проблеми при обробці природної мови:

- неоднозначність кордонів слова. На перший погляд може здатися, що токенізація являється простим завданням, але це не так, особливо при розробці багатомовного індексатора. У цифровій формі тексти деяких мов, таких, як китайський або японський, представляють складну задачу, тому що слова чітко не розділені пропуском. Мета токенізації в тому, щоб розпізнати

слова, які будуть шукати користувачі. Специфічна для кожної мови логіка використовується, щоб правильно розпізнати межі слів, що необхідно для розробки синтаксичного аналізатора для кожного підтримуваного мови [19];

- неоднозначність мови. Для більш точного ранжування документів пошукові системи можуть враховувати додаткову інформацію про слова, наприклад, до якої мови або частини мови воно належить. Ці методи залежать від мови, оскільки синтаксис між мовами різниться. При токенизації деякі пошукові системи намагаються автоматично визначити мову документа;

- різні формати файлів. Для того, щоб правильно визначити, які байти представляють символи документа, формат файлу повинен бути правильно оброблений. Пошукові системи, які підтримують різні формати файлів, повинні правильно відкривати документ, отримувати доступ до документа і токенизувати його символи;

- помилки пам'яті. Якість даних природної мови не завжди може бути досконалим. Уразливість існує через невідомого кількості документів, зокрема, в Інтернеті, які не підкоряються відповідного протоколу файлу. Двійкові символи можуть бути помилково закодовані в різних частинах документа. Без розпізнавання цих символів і відповідної обробки може погіршитися якість індексу або індексування [19].

На відміну від більшості людей, комп'ютери не розуміють структуру документа природної мови і не можуть автоматично розпізнавати слова і пропозиції. Для комп'ютера документ - це тільки послідовність байтів. Комп'ютер не «знає», що символ пробілу являється роздільником слів в документі. Людина повинна запрограмувати комп'ютер так, щоб визначити, що являється окремим словом, званим токеном. Таку програму зазвичай називають токенизатором або синтаксичним аналізатором, а також лексичним аналізатором.

Під час токенизації синтаксичний аналізатор визначає послідовність символів, які представляють слова та інші елементи, наприклад, пунктуація,

представлена числовими кодами, деякі з яких являються недрукованих керуючими символами. Синтаксичний аналізатор може розпізнати деякі об'єкти. При розпізнаванні кожного токена можуть бути збережені деякі характеристики, наприклад, мова або кодування, частина мови, позиція, число пропозиції, позиція в реченні, довжина і номер рядка [19].

Якщо пошукова система підтримує декілька мов, то першим кроком під час токенизації буде визначення мови кожного документа, оскільки багато подальші кроки залежать від цього.

Аналіз формату документа. Якщо пошукова система підтримує безліч форматів документів, то документи повинні бути підготовлені для токенизації. Проблема полягає в тому, що деякі формати документів містять інформацію про форматування на додаток до текстового змісту. Аналіз формату також згадується як структурний аналіз, поділ тегів, текстова нормалізація.

Аналіз формату може включати методи підвищення якості, щоб уникнути включення непотрібної інформації в індекс. Контент може керувати інформацією про форматування, щоб включати додаткові відомості [19].

3.2 Визначення пошукових систем

На даний час існує велика кількість пошукових систем та баз даних з методами повнотекстового пошуку. Для аналізу було взято пошукові системи:

- mySQL;
- postgreSQL;
- xapian;
- sphinx;
- solr;
- clucene.

Особливостями повнотекстового пошуку MySQL являються:

- індекси, що створюються тільки на полях MyISAM-таблиць;
- розмір індексу дорівнює половина від розміру даних, але так як відключити збереження текстів в базі даних неможливо, то розмір індексу збільшується до 150% від розміру даних [20];
- швидкість індексації без використання додаткових технологій дорівнює 1.5 МБ / с. Якщо у якості джерела даних використовувати базу даних та окремо розташований стеммер то швидкість індексації дорівнює 314 КБ / с;
- доступні англійські стоп слова;
- за замовчуванням логічний пошук йде з використанням операцій «АБО», тому для пошуку з операціями «І» потрібно кожне слово перетворювати в «+ слово»;
- має два режими пошуку – логічний і звичайний («natural language»). Логічний пошук перевіряє, чи існують слова в документі і підтримує логічні операції, фрази і префіксний пошук, але не повертає оцінку релевантності. Звичайний пошук підтримує релевантність, але не підтримує оператори пошуку;
- має префіксний пошук [20].

Особливостями повнотекстового пошуку PostgreSQL Textsearch являються:

- для індексування використовується індекс інвертований індекс на основі узагальненого інвертованого індексу(GIN), що має іншу назву Tsearch2, створений Олегом Бартуновим і Федором Сигаєва;
- має вбудовані стеммери;
- має підтримку стоп-слів, синонімів, тезауруса, словників ISpell;
- має можливість при індексації додати до кожної лексеми «вагу»;
- має окремі функції для ранжирування результатів і підсвічування шуканих слів у цитатах;
- має індексований тип даних, що називається text search vector;

- дозволяє створювати функціональні індекси. Для оптимізації дискового простору, можна не зберігати повний текст документів в індексі [20];

- має оператори пошуку – AND, OR і NOT і префіксний пошук;
- відсутні – пошук прилеглих слів, точних форм, фраз;
- розмір індексу – 150% від розміру даних, якщо самі тексти не зберігати, а зберігати тільки text search vector'и;

- швидкість індексації - 1.5 МБ / с, зі збільшенням об'єму даних швидкість індексації падає.

Особливостями пошукової системи Xapian являються:

- представлений у вигляді бібліотеки;
- має інтерфейси для різних мов: C ++, Java, Perl, Python, PHP, Tcl, C #, Ruby, Lua;
- для індексації використовується інвертований індекс на основі В-дерева;
- має стеммери для 15 мов;
- має підтримку стоп-слів і синонімів;
- може виправляти помилки без використання словників, а лише на основі індексованих даних [20];
- має «Faceted Search» – підрахунок агрегатних значень за всіма або майже всіма знайденими документами. Розмір індексу – більше за розмір даних в 2 рази.

Особливостями пошукової системи Sphinx являються:

- має окремий пошуковий сервер;
- має інтерфейсні бібліотеки для багатьох мов програмування;
- видалення проводиться установкою флагу «старий запис», і подальшим її видаленням з усіх результатів пошуку кожен раз;
- має сервер пошуку, з яким можна спілкуватися за власним протоколом або протоколом MySQL, розмір індексу – 1/3 даних. Може збільшитися в 2 рази, якщо включити індексацію точних словоформ;

- має вбудовані російський, англійський та чеський стеммери і препроцесори Soundex і Metaphone (для порівняння англійських слів за звучанням);

- стеммери для інших мов можна підключити додатково;
- має підтримку стоп-слів;
- має інфіксийний пошук;
- вміє індексувати по абзацах і у вмісті заздалегідь заданих HTML-тегів;

- може підсвічувати шукані слова в цитатах [20];

- швидкість індексації – 4.5 МБ / с.

Особливостями пошукової системи Solr (Lucene) являються:

- побудована на основі Lucene. Lucene – це Java-бібліотека пошуку, протестувати її повністю дуже складно - система найпотужніша з усіх розглянутих;

- являє собою пошуковий сервер, реалізований у вигляді веб-сервісу з XML / JSON / CSV-інтерфейсами. Для запуску вимагає сервлет-контейнер – Tomcat, або Jetty;

- інкрементальна індексація така ж швидка, як і індексація безлічі документів разом;

- розмір індексу 20-30% від розміру даних;

- має 31 вбудований стеммер;

- має різні аналізатори, що обробляють звучання, аббревіатури, стоп-слова, синоніми, «protect-слова», словосполучення, шингли, поділ слів, URL;

- має реплікацію індексів, автоматичну кластеризацію результатів пошуку (Carrot2), пошукового робота (Nutch), підтримку розбору бінарних документів (Word, PDF) за допомогою Tika;

- швидкість індексації – 2.75 МБ / с;

Особливостями CLucene являються:

- повільний розвиток;

- мало інтерфейсів;

- застаріла документація;
- являється одним із портів Lucene;
- швидкість індексації – 3.8 МБ / с;
- розмір індексу 20-30% від розміру даних.

Таблиця 3.1 – Аналіз пошукових систем

	MySQL	PostgreSQL	Xapian	Sphinx	Solr	CLucene
Швидкість індексації	314 КБ/с	522 КБ/с	1.36 МБ/с	4.5 МБ/с	2.75 МБ/с	3.8 МБ/с
Швидкість пошуку	175 мс / 3.46 сек	28 мс / 2.1 сек	14 мс / 135 мс	7 мс / 75 мс	25 мс / 212 мс	10 мс / 212 мс
Розмір індексу	150%	150%	200%	30%	20%	20%
Реалізація	СУБД	СУБД	Бібліотека	Сервер	Сервер	Бібліотека
Кількість інтерфейсів	∇	∇	9	6+∇	8	3.5
Оператори пошуку	&*"	&*	&*N~	&*N- <Z	&*N- ~	&*N~
Стемери	-	15	15	15	31	15+CJK
Стоп-слова, синоніми	-	+	+	+	+	+
Soundex	-	-	-	+	+	+

Для аналізу методів пошуку в пошукових системах були використані наступні умовні позначення:

- & - булеві операції;
- * - префіксний пошук;
- " - точна фраза;
- N - слова поблизу;
- - - діапазони;

- ~ - приблизний пошук;
- < - порядок слів;
- Z – зони.

Виходячи з отриманих даних (таблиця 3.1) було виявлено, що найбільш оптимальна пошукова система для електронних бібліотек являється Sphinx через найбільшу швидкість індексації та найменший час пошуку.

3.3 Реалізація модифікованого методу пошуку

Хоч методи пошуку мають різний функціонал, але він не покриває всі потреби користувача. Одною із таких потреб являється можливість додавати транслітерацію слів в пошуковому запиті.

Для розрахунку часу виконання сформованого запиту без урахування оптимізації запитів та кешування даних, слід використовувати формулу:

$$t = (n+1) \cdot t_{cp}, \quad (3.1)$$

де t_{cp} – середній час виконання запиту;

n – кількість додаткових опцій.

Для додання можливості використання різних транслітерацій у пошукового запиту було зроблено модифікацію не метод пошуку. Модифікація виступає у вигляді проміжного рівня між користувачем та пошуковою системою.

При тестуванні модифікованого методу пошуку з пошуковою системою Sphinx були отримані практичні значення часу виконання запиту. Це дозволяє порівняти теоретичні розрахунки для немодифікованого методу з практичними даними модифікованого методу, що наведені на таблиці 3.2.

Таблиця 3.2 – Аналіз ефективності модифікованого методу пошуку

	n=0	n=1	n=2	n=3	n=4	n=5
Результати не модифікованого методу пошуку	7 мс	14 мс	21 мс	28 мс	35 мс	42 мс
Результати модифікованого методу пошуку	7 мс	12 мс	18 мс	22 мс	27 мс	31 мс

Виходячи з отриманих даних (таблиця 3.2) можна стверджувати, що розроблена модифікація методу підвищує ефективність виконання пошукових запитів запитів. Нижче наведено результати розрахунків у вигляді діаграми (рисунок 3.1).

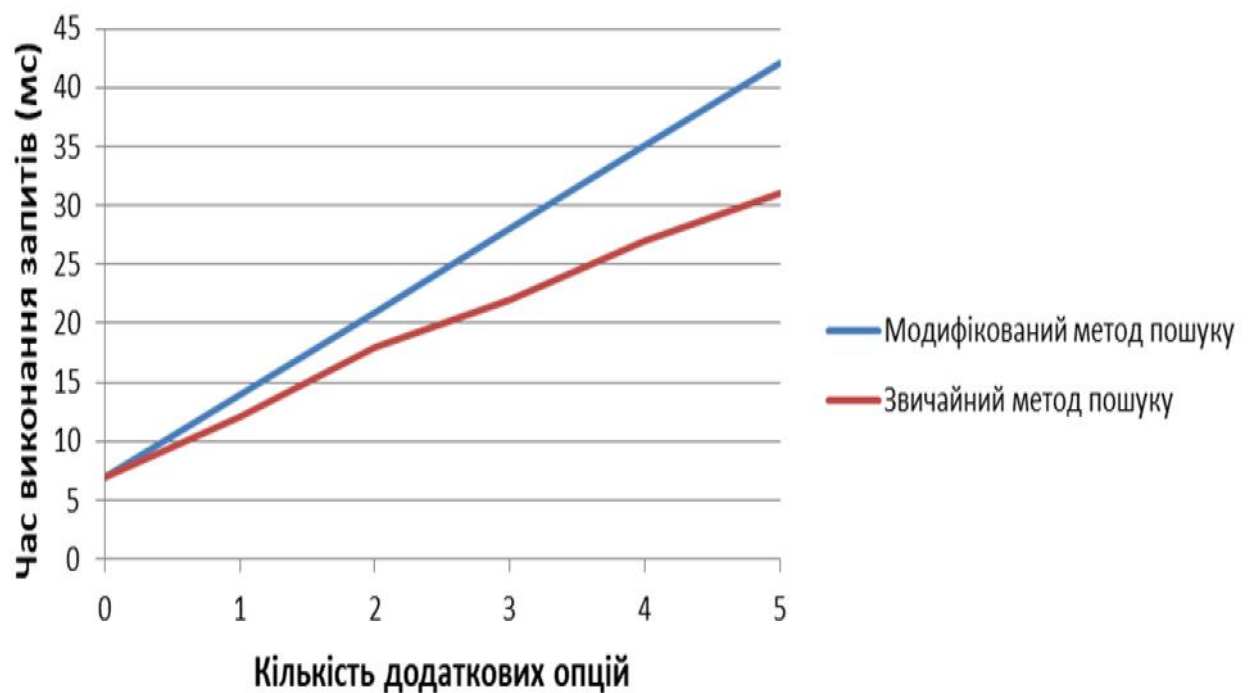


Рисунок 3.1 – Діаграма результатів аналізу ефективності модифікованого методу

На діаграмі (рисунок 3.1) видно, що модифікація підтверджує свою ефективність на будь-якій кількості додаткових опцій. Блок схему алгоритму модифікації пошукового методу наведено на рисунку 3.2.

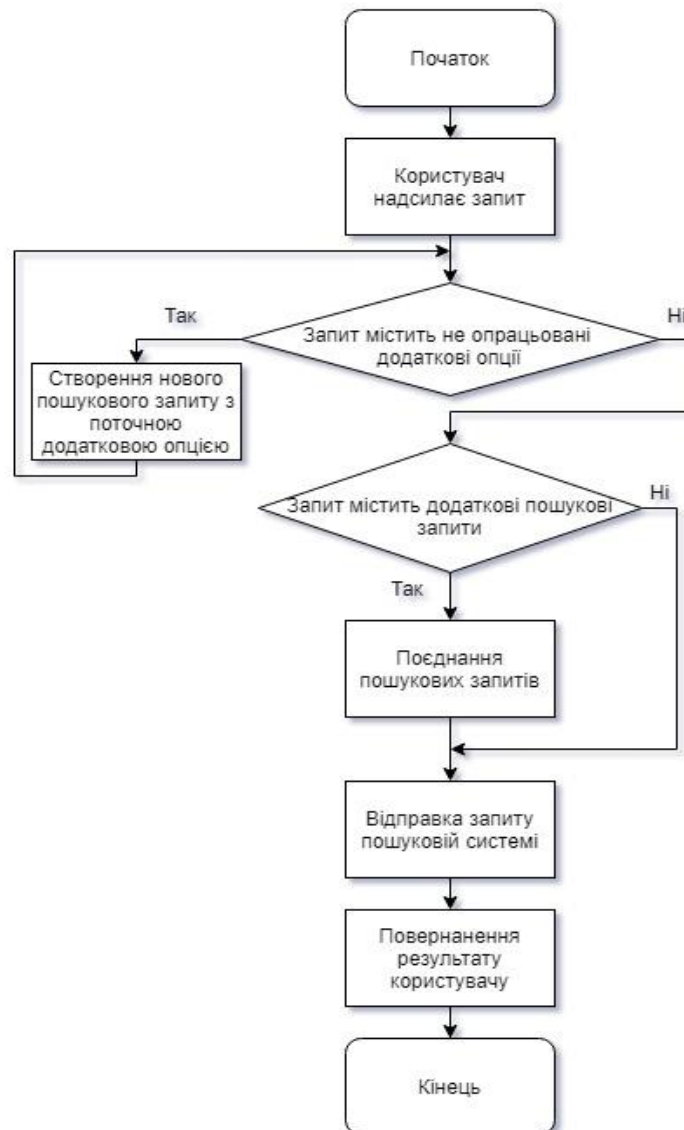


Рисунок 3.2 – Блок-схема алгоритму модифікації методу пошуку

Користувач зі своєї сторони надає пошуковий запит та спеціалізує інші транслітерації або синоніми до бажаних слів. Проміжний слой перетворює один запит до пошукової системи в декілька в залежності від вказаних транслітерацій та передає запити до пошукової системи.

Після закінчення пошуку проміжний рівень формує результат викидаючи дублікати, сформовані зарахунок декількох запитів.

Для підвищення швидкості виконання метода, сформовані запити поєднуються до одного комплексного запиту, що дозволяє використовувати вбудовану у пошукову систему та базу даних оптимізацію.

Основною перевагою цієї модифікації є оптимізація запитів та автоматизація процесу обробки дублікатів, що підвищує швидкість пошуку в порівнянні зі звичайним пошуком та ручним аналізом. Також зменшиться час виконання запитів за рахунок більш оптимізованих запитів. При цьому пропорційно кількості доданих умов збільшиться час виконання запиту.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА

Інформаційна система для персональної електронної бібліотеки створена орієнтуючись в першу чергу на читачів. Вона складається з 4 проектів (рисунок 4.1):

- Core;
- Services;
- Infrastructure;
- WindowsDesktopUI.

Проект Core містить опис об'єктів, що використовуються у програмному забезпеченні.

Проект Infrastructure містить логіку зв'язку з базою даних та пошуковою системою.

Проект Services містить логіку програми та поєднує рівень користувача з рівнем даних. Також на цьому рівні знаходиться розроблена модифікація методу пошуку.

Проект WindowsDesktopUI містить реалізацію інтерфейсу програми для операційної системи Windows та логіку взаємодії з сервісами.

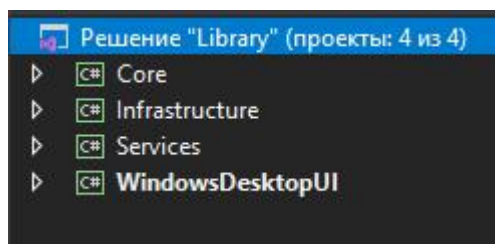


Рисунок 4.1 – Структура інформаційної системи

Перейдемо до інтерфейсу програми. Головне вікно зображено на рисунку 4.2.

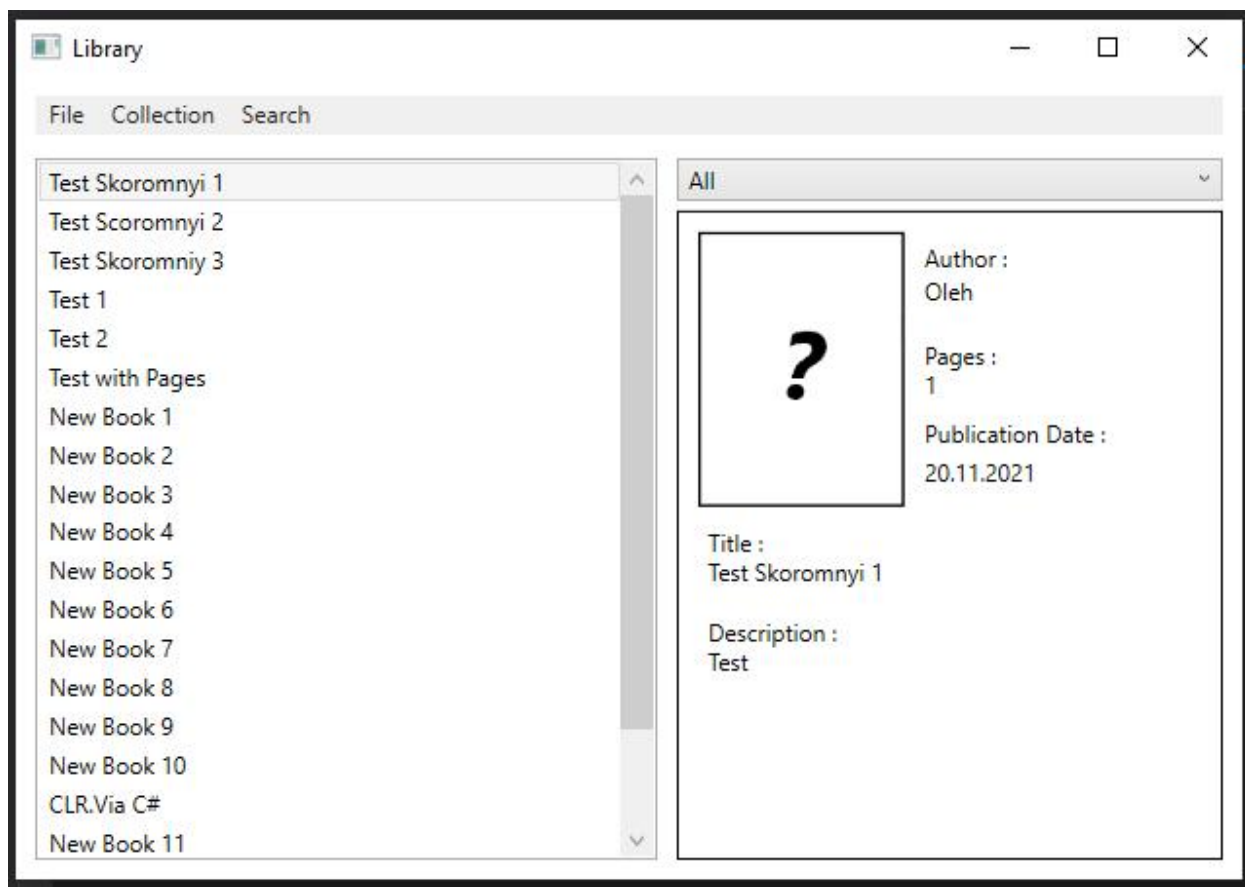


Рисунок 4.2 – Головне вікно програми

На головному вікні (рисунок 4.2) знаходяться такі елементи, як:

- список документів;
- перемикач колекцій;
- елемент меню “File”;
- елемент меню “Collection”;
- елемент меню “Search”;
- кнопки керування вікном;
- назва вікна;
- відсік, що містить інформацію про обраний документ (зображення, авторів, кількість сторінок, дату публікації, назву та опис).

Для відкриття документу, слід зробити подвійне натискання лівої кнопки миші по бажаному документу в списку. Після цього вікно зміниться на нове. Нове вікно (рисунок 4.3) містить у собі:

- кнопку повернення до початкового вікна (Back);
- кнопку зміни кольорової схеми (Night Mode);
- слайдер для керування позицією у документі;
- відображення кількості сторінок;
- відображення поточної сторінки з можливістю її редагування;
- назва вікна з назвою обраного документу;
- відображення тексту документу.

Для зміни кольорової схеми слід натиснути кнопку “Night Mode”. Для повернення на сторінку бібліотеки слід натиснути кнопку “Back”.

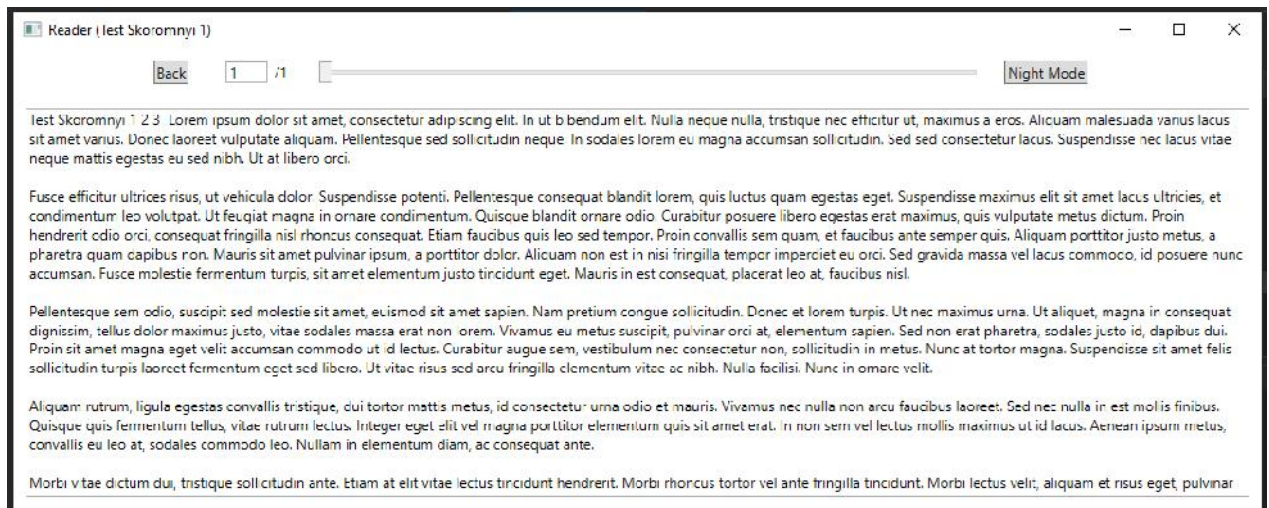


Рисунок 4.3 – Вікно читання документів

Для відкриття пошуку документів, слід натиснути на пункт меню “Search” на головному вікні. Після цього вікно зміниться на нове. Нове вікно (рисунок 4.4) містить у собі:

- кнопку пошуку;
- кнопку розкриття додаткових опцій;
- кнопку додання опції пошуку;
- текстове поле пошукового запиту;
- пари текстових полів для опцій;
- назва вікна;

- відображення знайдених документів.

Для того, щоб здійснити пошук слід написати до текстового поля пошуковий запит та натиснути кнопку пошуку. Знайдені результати будуть наведені у списку нижче. Для відкриття документу слід зробити подвійний клік лівою кнопкою миші по бажаному документу.

У разі, коли потрібно додати додаткові опції пошуку, наприклад, транслітерацію або синонім, то потрібно спочатку розкрити додаткові опції пошуку. Потім натиснути кнопку “Add option”. З’явиться пара текстових полей. До першого поля заноситься слово з пошукового запиту, що має різні варіанти. До другого поля заноситься інший варіант слова з першого поля.

Після цього слід натиснути кнопку пошуку, або, у разі необхідності додати додаткових опцій.

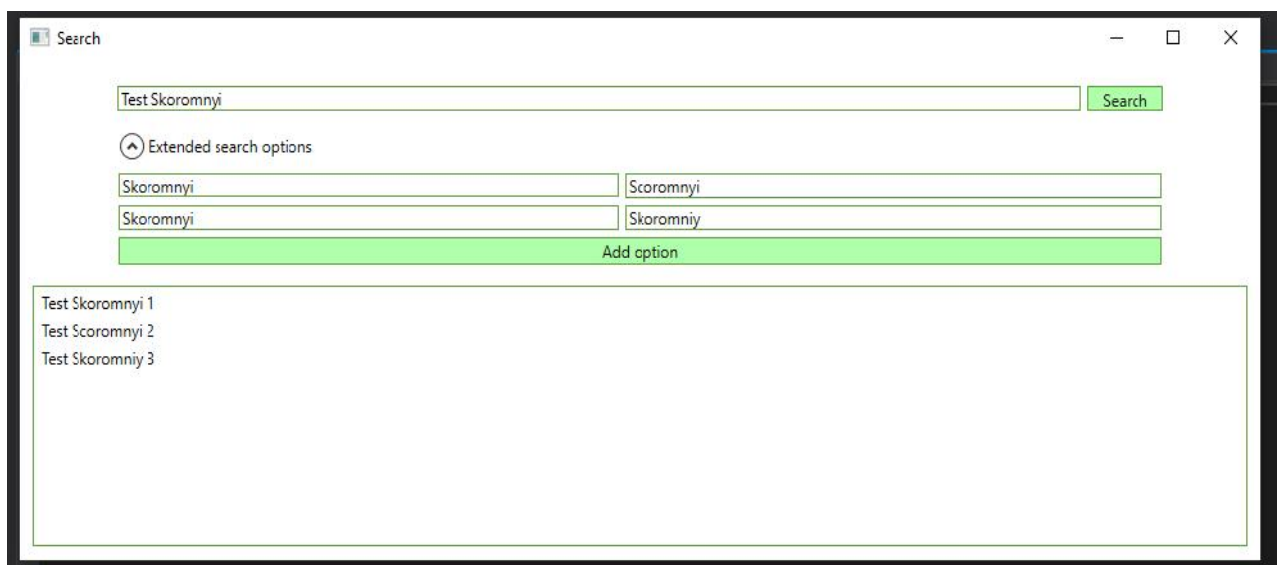


Рисунок 4.4 – Вікно пошуку документів

Виходячи з наведеної інструкції користувача можна підсумувати, що програмний застосунок не виявиться складним для нових користувачів.

ВИСНОВКИ

За результатами роботи було розроблено інформаційну систему персональної багатофункціональної електронної бібліотеки для персонального наповнення електронними документами.

В роботі було проведено аналіз різних підходів що до реалізації завдання з використання різних платформ та середовищ. В ході досліджень була обрана платформа.NET Core на мові C# в середовищі Visual Studio з використанням WPF для побудови графічного інтерфейсу.

Було проведено аналіз методів та алгоритмів повнотекстового пошуку та розроблено і побудовано модель кросплатформенного програмного забезпечення для створення інформаційної системи.

За результатами проведеного аналізу та досліджень було створено спеціалізований програмний продукт, що призначений для індивідуального наповнення, організації власного користування персональною електронною бібліотекою читачами. Створена персональна бібліотека має особисте заповнення документами, зручний інтерфейс та потужний пошук.

Наукова новизна результатів роботи полягає в модифікації методу пошуку в інформаційних системах персональних електронних бібліотек при використанні синонімів та транслітерацій за рахунок додання проміжного рівня між користувачем та пошуковою системою.

Практичне значення отриманих результатів полягає в запропонованих технологічних рішеннях удосконалення методів пошуку та процесів розробки інформаційних систем, які можуть бути використані розробниками програмного забезпечення при розробці власних рішень подібного спрямування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 7448:2013. Бібліотечно-інформаційна діяльність. Терміни та визначення понять. [Чинний від 2013-11-29]. Вид. офіц. Київ, 2014. – 45 с.
2. Хвостенко И. М. Информационные технологии в библиотеках: дайджест-конспект / И. М. Хвостенко, Н. П. Носова, В. Г. Деев. – Новосибирск: НГОНБ, 2015. – 90 с.
3. Вітенко В. І. Технологія створення електронної бібліотеки: підходи та перспективи / В. І. Вітенко. – Кіровоград, 2003. – 7 с.
4. Пилко И. С. Информационные и библиотечные технологии: учеб. пособие / И. С. Пилко – СПб. : Профессия, 2008. – 342 с.
5. Андрухів А. І. Електронні послуги в бібліотечній справі: навч. посіб. / Андрій Андрухів, Роман Голощук, Марія Сокіл. – Львів : Видавництво Львівської політехніки, 2016. – 136 с.
6. Назаровець С. Бібліотека 4.0: технології та сервіси майбутнього: [передові інтернет-технології для впровадження інноваційних бібліотечних послуг і сервісів, скерованих на задоволення потреб користувачів] / С. Назаровець, Є. Кулик // Бібліотечний вісник. – 2017. – № 5. – С. 3–14.
7. Лапо П. М. Введение в электронные библиотеки / П. М. Лапо, А. В. Соколов., 2005. – 90 с.
8. Грофф Д. SQL: полное руководство. 3-е издание. / Д. Грофф, П. Вайнберг, Э. Оппель. – Москва: Издательский дом "Вильямс", 2015. – 960 с.
9. Бондарь А.Г. Microsoft SQL Server 2014 – СПб.: БХВ-Петербург, 2015 – 592 с.
10. Juba S. Learning PostgreSQL 10 - Second Edition / S. Juba, A. Volkov., 2017. – 488 с.
11. Брэдшоу Ш. Mongo DB. Полное руководство / Ш. Брэдшоу, Й. Брэзил, К. Ходоров. – Москва: ДМК Пресс, 2020. – 540 с.
12. LiteDB v5 – A .NET NoSQL Document Store in a single data file

[Електронний ресурс] – Режим доступу: <https://www.litedb.org/docs/> – 01.12.2021 г. – Загол. з екрану.

13. Фленов М. Библия C# / Михаил Фленов. – Санкт-Петербург: БХВ-Петербург, 2020. – 512 с.

14. Сьерра К. Изучаем Java. Мировой компьютерный бестселлер / К. Сьерра, Б. Бейтс. – Москва: ЭКСМО, 2012. – 605 с.

15. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 [Текст] / Эндрю Троелсен. - М.: Вильямс, 2015. – 486 с.

16. Johnson B. Professional Visual Studio 2012 / Bruce Johnson. – Great Britain: Wrox, 2012. – 1104 с.

17. Колисниченко Д. Н. Поисковые системы и продвижение сайтов в Интернете. – М.: Диалектика, 2007. – 272 с.

18. Sergey Brin, Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine – Stanford University, Stanford: Computer Science Department, 2006. – 11 с.

19. Шемаєва Г. Перспективні напрями розвитку електронних інформаційних ресурсів бібліотек / Г. Шемаєва // Бібліотечний вісник – 2012. – № 3. – С. 3–7.

20. Носик А.М., Скоромний О.Г. Аналіз методів пошуку інформації для персональних електронних бібліотек. Проблеми інформатизації : зб. тез доп. IX міжнар. наук.-практ. конф. том 2, м. Харків, 18-19 листопада 2021 р. Харків, 2021. С. 86.