

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних систем та автоматики
Кафедра комп'ютерних систем управління
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
Освітньо-професійна програма Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ
Дубовой В.М.

«___» _____ 2019 року

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

Інформаційна система ієрархічної категоризації науковців за відкритими даними з Google Scholar. Частина 2. Програмна реалізація системи та експериментальні дослідження.

08-01.МКР.007.00.000

Студент групи 2АКІТ-18м Білоус Д.Д.

Керівник д.т.н., професор Штовба С.Д.

Рецензент к.т.н., доцент Софіна О.Ю.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних систем та автоматики
 Кафедра комп'ютерних систем управління
 Освітньо-кваліфікаційний рівень магістр
 Спеціальність 151 Автоматизація та комп'ютерно-інтегровані
 Освітньо-професійна програма Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ
В.М. Дубовой

“ 02 ” 09 2019 року
 Протокол №1 засідання кафедри КСУ від 02.09.2019

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Білоусу Денису Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Інформаційна система ієрархічної категоризації науковців за відкритими даними з Google Scholar. Частина 2. Програмна реалізація системи та експериментальні дослідження.»
 керівник магістерської кваліфікаційної роботи Штовба Сергій Дмитрович, д. т. н., професор.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “02” 10 2019 року № 254

2. Строк подання студентом магістерської кваліфікаційної роботи 12.12.2019 р.

3. Вихідні дані до магістерської кваліфікаційної роботи
підтримка браузерів Firefox, Chrome, Safari, IE810-11-Microsoft Edge., підтримка ОС – Windows , Android; максимальний час завантаження – 5 с; максимальна кількість запитів до системи – до 1000 шт/с; мови графічного інтерфейсу –, англійська, робота на більшості мобільних пристроїв..

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) – Вступ, огляд стану питання та постановка задачі, розробка програмного забезпечення, веб-системи ієрархічної категоризації науковців, тестування та експериментальні дослідження, економічна частина.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Постановка задачі дослідження, актуальність теми, UML-діаграма варіантів використання, UML-діаграма діяльності, UML-діаграма послідовності, інтерфейс системи, інтерфейс модального вікна веб-системи, тестування пошуку науковця тестування пошуку науковця, модальне вікно, тестування пошуку за галузями, тестування рейтингування науковців

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	д.т.н., професор кафедри КСУ Штовба С.Д.		
4	к.т.н., доцент кафедри ЕПВМ Ратушняк О.Г.		

Календарний план

№ з/п	Назва етапів Роботи	Строк виконання етапів роботи	Примітка
1	Огляд стану питання та постановка задач дослідження	07.10.2019р.	
2	Аналіз функцій системи та розробка архітектури системи	15.10.2019р.	
3	Аналіз структури системи та розробка структурної схеми системи	22.10.2019р.	
4	Проектування інтерфейсу системи	29.10.2019р.	
5	Розробка програмного забезпечення системи	05.11.2019р.	
6	Тестування програмного забезпечення	12.11.2019р.	
7	Економічна частина	19.11.2019р.	
8	Оформлення пояснювальної записки, графічного матеріалу і презентації	25.11.2019р.	
9	Захист МКР	12.12.2019р.	

7. Дата видачі завдання “_05_”_09_2019 року

Студент _____ Білоус Д.Д.

Керівник магістерської кваліфікаційної роботи _____ Штовба С.Д.

ЗМІСТ

АНОТАЦІЯ	6
ABSTRACT	7
ВСТУП.....	8
1 ОГЛЯД СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1. Інформаційно-пошукові веб-системи як об’єкт проектування	10
1.2. Формалізація вимог до веб-системи ієрархічної категоризації науковців	13
1.3. Варіативний аналіз технологій створення інформаційно-пошукових веб-систем	15
1.3.1. Аналіз сучасних вимог до реалізації веб-системи.....	15
1.3.2. Аналіз сучасних підходів до реалізації системи.....	25
1.3.3. Аналіз сучасних засобів реалізації системи	27
1.4. Задачі дослідження.....	45
1.5. Висновки до розділу.....	46
2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-СИСТЕМИ ІЄРАРХІЧНОЇ КАТЕГОРИЗАЦІЇ НАУКОВЦІВ.....	48
2.1. Розробка архітектури системи	48
2.1.1. Визначення функцій системи.....	48
2.1.2. Розробка алгоритму функціонування системи	52
2.2. Розробка структури системи	56
2.2.1. Модуль інтерфейсу системи	59
2.2.2. Модуль пошуку	60
2.2.3. Модуль сортування.....	61
2.2.4. Модуль фільтрів	62

	5
2.3. Проектування інтерфейсу	63
2.3.1. Проектування інтерфейсу веб-сторінок системи.....	63
2.3.2. Дослідження математичних підходів до визначення показників usability веб-системи	70
2.4. Висновки до розділу.....	74
3 ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	76
3.1. Пошук науковця за прізвищем.....	76
3.2. Пошук науковців за галузями	78
3.3. Пошук науковців за спеціальностями	80
3.4. Рейтингування науковців.....	81
3.5. Дослідження розподілів науковців	84
3.5.1. Розподіли за цитованістю.....	85
3.5.2. Розподіли за галузями.....	87
3.5.3. Розподіли за Хіршем.....	90
3.6. Висновки до розділу.....	91
4 ЕКОНОМІЧНА ЧАСТИНА.....	93
4.1. Оцінювання комерційного потенціалу розробки.....	93
4.2. Прогнозування витрат на виконання науково-дослідної роботи	97
4.3. Оцінка внеску НДР.....	101
4.4. Висновок.....	104
ВИСНОВКИ.....	105
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	106
ДОДАТКИ.....	111
Додаток А	112
Додаток Б.....	115
Додаток В	128

АНОТАЦІЯ

Розроблена інформаційно-пошукова веб-система на основі ієрархічної категоризації науковців. Початковою інформацією для наповнення системи є профілі науковців в Google Scholar. На основі інтересів в профілях здійснено категоризацію науковців за системою класифікації наук ANZSRC. Веб-система надає такі сервіси: пошук науковця за іменем, формування списку науковців з відповідної спеціальності або спеціальностей, ранжування обраних науковців за індексом цитування, індексом Хірша та і10-кількістю статей, що мають 10 та більше цитувань. Систему реалізовано у відповідності до сучасних потреб та вимог які ставляться перед веб-системами.

ABSTRACT

Web search engine based on hierarchical categorization of scientists was developed. The initial information to fill the system is Google Scholar Scholar Profiles. Based on the interests in the profiles, the categorization of scientists according to the ANZSRC science classification system was carried out. The Web system provides the following services: search for a scientist by name, form a list of scientists in the relevant specialty or specialties, ranking selected scientists by citation index, Hirsch index and i10-number of articles with 10 or more citations. The system is implemented in accordance with modern needs and requirements that are put before web systems.

ВСТУП

В сучасному світі люди повсякчас стикаються із задачею пошуку інформації. Наразі більша частина людства постійно користується мережею Інтернет для пошуку тієї чи іншої інформації. Можливість пошуку в мережі Інтернет забезпечують інформаційно-пошукові системи найбільш очевидними прикладами яких можна назвати зокрема такі сервіси як Yahoo та Google. Більшість таких систем відповідають за пошук широкого набору інформації а також існують системи які дозволяють здійснювати пошук більш спеціалізованої інформації. Однак на даний момент немає систем які дозволяли б здійснювати пошук науковців які є класифікованими за певною галуззю чи підгалуззями і при цьому надавали широкий набір інформації про кожного науковця. Подібним питанням займається бібліометрика української науки – єдине джерело українських науковців, що класифіковані за галуззями науки. Проте в існуючій системі пошук дозволяє отримати невелику кількість про науковця, крім того в ній проблема задача класифікації вирішується суб'єктивно, тобто, класифікація науковця виконується людиною. Окрім цього відношення до галузі науки одиничне хоча науковець може бути обізнаним у декількох галузях науки.

Розробка інформаційної системи дозволить проводити пошук науковців за галуззями науки, проводити їхнє рейтингування та автоматизувати трудомісткий процес класифікації.

Мета дослідження: підвищення ефективності та якості процесу пошуку категоризованих науковців шляхом розробки інформаційної веб-системи ієрархічної категоризації науковців.

Об'єкт дослідження: інформаційні системи підтримки наукової діяльності

Предмет дослідження: інформаційна технологія ієрархічної категоризації науковців на основі їх інтересів.

Методи дослідження: У дослідженнях використовувались методи UML моделювання для розробки архітектури системи; методи реалізації адаптивних веб-систем для розробки інтерфейсу інформаційної системи.

Наукова новизна отриманих результатів:

1. Розроблена веб-система ієрархічної категоризації науковців;
2. Вперше розроблено інформаційну веб-систему яка дозволяє здійснювати багатофункціональний пошук категоризованих науковців;
2. Вперше запропоновано систему яка дозволяє відображати показники належності науковців до галузей науки.

Практичне значення одержаних результатів полягає у можливості використання розробленої інформаційної системи для пошуку науковців, пошуку науковців за галуззю, рейтингування науковців можливості в подальшому застосуванні для складання зайнятості наукової спільноти, факультету тощо.

Публікації. тези дослідження подано на конференцію «Молодь в науці: дослідження, проблеми, перспективи»

1 ОГЛЯД СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Інформаційно-пошукові веб-системи як об'єкт проектування

Пошук інформації є невід'ємною складовою життя сучасної людини. Зі збільшенням кількості джерел звідки можна цю інформацію отримати актуальність систем які дозволяють знайти те що потрібно користувачу лише зростає. Пошук будь-якої інформації в мережі інтернет зараз здійснюється з допомогою різного роду інформаційно-пошукових систем.

Інформаційно-пошукові системи – це автоматизовані системи, призначенні для збору, пошуку, обробки, збереження, та видачі інформації [1].

Такі системи позбавляють необхідності в перегляді великої кількості інформації користувачем і надає лише ту інформацію яка відповідає пошуковому запиту. Інформаційно-пошукові системи в залежності від свого призначення можуть надавати доступ до книг, журналів чи до іншої інформації. Якщо брати до уваги принцип дії інформаційно-пошукової системи то можна виділити такі її види як пошукові тематичні каталоги, рейтингові системи і індексні пошукові системи. Однак на даний момент найбільш поширені індексні пошукові системи, до яких відносяться і зокрема пошукові системи(машини) такі як ресурси Google, Yahoo, Yandex. Пошукові машини є найбільш очевидним застосуванням інформаційно пошукових систем [2].

Окрім того існує багато інших веб-ресурсів які дозволяють шукати більш специфічну інформацію.

Інформаційно-пошукова система складається з двох компонентів: системи індексування і системи запитів. Перша з них відповідає за аналіз документів завантажених з всесвітньої павутини і за створення індексів які дозволять в подальшому виконати пошукові запити. Друга це видимий інтерфейс пошукової машини тобто частина з якою взаємодіє користувач [3].

Для того щоб здійснити пошук у базі даних системи створюється певний опис для кожного документу чи іншого представлення інформації. Такий опис традиційно називають пошуковим образом документа.

Пошуковий образ документа (ПОД) – це текст інформаційно-пошуковою мовою, поставлений в однозначну відповідність документу, що відображає його ознаки, необхідні для пошуку його за запитом. ПОД містить різноманітні дані про документ такі як ключові слова, що це за документ та все інше що дозволяє знайти конкретний документ у відповідності до пошукового запиту [4].

В більш сучасному розумінні інформаційно-пошукових систем використовують такий термін як індекс. Під індексом розуміють масив інформації який містить інформацію про документ та посилання на нього. Процес створення таких індексів називають індексуванням. Індокси зберігаються в базі даних і коли користувач здійснює пошук система звертається до даної бази даних і надає в результатах документи індекси яких відповідають пошуковому запиту. Для того щоб система могла зрозуміти пошуковий запит користувача існують різноманітні інформаційно пошукові мови які дозволяють перетворити запит у зрозумілу для системи форму.

Інформаційно-пошукові мови — це синтетично створені мови для ідентифікації і пошуку документів за запитом [5].

Існують різні інформаційно-пошукові системи кожна з яких відрізняється від іншої певними властивостями однак дані системи мають наступні спільні процедури якщо справа стосується збору, зберігання і видачі інформації:

- підбирання і аналіз документів
- створення пошукового образу документів(індексування);
- запис документів і індексів до бази даних;
- зберігання документів і індексів;
- аналіз пошукових запитів;
- видача документів користувачам.

Якщо брати до уваги відмінності то інформаційно-пошукові системи можна класифікувати за наступними ознаками:

- режимом пошуку;

- типом інформаційно-пошукової мови;
- родом виконуваних операцій;
- типом критерію відповідності;
- ступенем автоматизації.

За режимом пошуку бувають системи вибіркового розподілу інформації і системи ретроспективного пошуку.

За типом інформаційно-пошукової мови системи класифікуються на системи з природними і системи з інформаційними мовами.

За родом виконуваних операцій виділяють документальні, фактографічні, логічні і комплексні інформаційно-пошукові системи.

- документальні інформаційно-пошукові системи у відповідь на пошуковий запит видають оригінали чи копії документів з необхідною інформацією;

- фактографічні інформаційно-пошукові системи у відповідь на пошуковий запит видають відповідні фактичні дані(формули, характеристики матеріалів і т.д.);

- логічні інформаційно-пошукові системи на пошуковий запит видають не лише інформацію але й за необхідності здійснюють певну логічну обробку цієї інформації для одержання нової, яка в систему явно не вводилася;

- комплексні ж інформаційно-пошукові системи містять сукупність всіх перерахованих вище типів систем. [5]

За критерієм відповідності, тобто сукупністю правил визначення ступеня формальної відповідності між ПОД і пошуковим розпорядженням ПР, інформаційно-пошукові системи поділяються на наступні:

- системи з використанням логічних схем;
- системи з використанням аналітичних функцій;
- системи з використанням «вагових» функцій чи коефіцієнтів;
- системи з аналізом критерію відповідності на збіг чи часткове входження. [5]

За ступенем автоматизації інформаційно-пошукові системи виділяють наступні види систем:

- системи з автоматичною класифікацією масивів;
- системи з автоматичним пошуком для сортування і порівняння ПОД з пошуковим розпорядженням;
- системи з автоматичним індексуванням для автоматизації процесів згортання при реферуванні, анотуванні, виборі ключових слів і т. ін.;
- системи з автоматичною видачею різновидів інформації, системи з автоматичним управлінням. [5]

Якщо брати до уваги принцип дії інформаційно-пошукової системи то можна виділити такі її види як пошукові тематичні каталоги, рейтингові системи і індексні пошукові системи. Однак на даний момент найбільш поширені індексні пошукові системи, до яких відносяться і зокрема пошукові системи(машини) такі як ресурси Google, Yahoo, Yandex.

Однак не зважаючи на існування різних типів інформаційно-пошукових систем всі вони покликані для того щоб у відповідь на пошуковий запит надавати релевантну інформацію.

1.2. Формалізація вимог до веб-системи ієрархічної категоризації науковців

Веб-система ієрархічної категоризація науковців програмна реалізація якої є темою даної роботи покликана надати підтримку в науковій діяльності науковцям. Базуючись на відкритих даних Google Scholar дана система містить певний набір інформації про українських науковців.

Головною метою системи є допомога науковцям в пошуку колег для співпраці в залежності від конкретної галузі чи підгалузі науки. Загалом веб-система надає різні сервіси які тим чи іншим способом дозволяють досягнути даної мети.

Початково система має представляти її користувачам перелік різних українських науковців про кожного з яких в результатах пошуку

представляється коротка інформація з допомогою якої вони можуть прийняти те чи інше рішення щодо подальших дій. Серед даної інформації можна виділити показники цитування конкретного науковця, індекс Хірша, i10-індекс та його наукові інтереси.

Користувач сервісу зможе здійснити пошук конкретного науковця якщо він знає його ім'я і далі в результатах пошуку знайти необхідну йому інформацію.

Якщо ж користувач не шукає конкретного науковця а його цікавлять науковці наукова діяльність яких зосереджена в конкретній галузі науки то вони зможуть відфільтрувати результати пошуку за конкретною галуззю чи підгалуззю науки з допомогою наявних фільтрів по галуззях. Відмітивши у фільтра певну галузь чи підгалузть науки користувач також зможе встановити певний ступінь належності науковця до неї. Окрім того користувач веб-системи зможе відфільтрувати результати пошуку за ключовими словами і наприклад таким чином знайти науковців з конкретними науковими інтересами.

Безумовно навіть відфільтрувавши науковців за галузями науки може бути багато зайвих результатів пошуку. Тому у веб-системі можна буде здійснити сортування за показниками цитування науковців в результаті чого система відобразить на початку тих науковців які мають найбільші значення в цьому аспекті.

Наявність функції розширеного сортування науковців дозволить здійснити більш детальніше сортування і наприклад таким чином відобразити результати пошуку за показниками цитування починаючи з 2013 року. Дана функція дозволить відобразити на початку результатів пошук науковців у яких за останні роки показник цитування є високим що свідчить про їх активну наукову діяльність.

Також веб-система дозволить переглянути більш детальну інформацію про окремого науковця. До вікна детальної інформації про науковця окрім імені та зображення буде входити наступна інформація:

- назва організації в якій працює науковець;
- посилання на профіль науковця в Google Scholar;

- наукові інтереси науковця ;
- показники цитування, індексу Хірша та i10-index за всі роки і окремо за роки починаючи з 2013 року;
- діаграма належності наукових робіт науковця до галузей науки;
- діаграма належності наукових робіт науковця до підгалузей науки.

Діаграми належності науковця до галузей чи підгалузей дозволять оцінити чи є наукова діяльність науковця у конкретній галузі основною чи можливо він робить більший акцент на діяльність в іншій галузі чи то підгалузі науки.

1.3. Варіативний аналіз технологій створення інформаційно-пошукових веб-систем

1.3.1. Аналіз сучасних вимог до реалізації веб-системи

Якою б не була система яку ви створюєте завжди важливо максимально пропрацювати реалізацію усіх її складових незважаючи на те чи це якесь технічне обладнання, чи програмне забезпечення чи web-ресурс. На сьогоднішній день при створенні web-ресурсу важливо не лише забезпечити ефективну обробку інформації ресурсом але й не менш важливо як ви в підсумку представляєте цю інформацію користувачеві.

Розробка web-ресурсу обов'язково повинна включати в себе розробку якісного дизайну ресурсу для того щоб забезпечити максимально якісне і зручне представлення інформації користувачеві. Сучасні технології невпинно розвиваються у зв'язку з чим з'являється велика кількість ресурсів на яких користувач зможе отримати інформацію яку він шукає. Разом із швидким ритмом сучасного життя велика кількість джерел інформації спонукають користувачів відсіювати багато непотрібної їм інформації. Тому користувачі схильні швидко покидати сайт якщо вони не можуть легко знайти на ньому потрібну інформацію. Щоб забезпечити максимально ефективно надання інформації ресурсом потрібно створити дизайн який зрозуміло і зручно цю інформацію відобразить. Так згідно з проведеними дослідженнями за

статистикою час за який користувач формує думку про ваш веб-сайт та вирішує залишитися на ньому чи покинути його складає 0,05 секунди (50 мілісекунд) [6]. Крім того у вас є близько 10 секунд щоб вразити користувача перш ніж він покине ваш сайт. Тому ваш веб-сайт повинен захопити увагу користувача протягом перших 10 секунд щоб зберегти його зацікавленість [7].

Незважаючи на відносно невеликий період існування мережі інтернет підхід до того як повинен виглядати web-ресурс стрімко розвивається. Тому в сучасному світі де люди мають вибір серед більше ніж мільярда веб-сайтів які вони можуть відвідати, вам потрібно переконатися що дизайн вашого сайту оптимізований у відповідності з юзабіліті (usability) та досвідом користування (user experience) [7].

Юзабіліті (англ. usability — дослівно «зручність користування») — поняття в мікроергономіці, що визначає загальну ступінь зручності предмета при використанні; термін схожий з терміном «ергономічність», але має іншу область поширення і відрізняється нюансами визначення. Використовується для комп'ютерної техніки, терміном юзабіліті називають концепцію розробки користувацьких інтерфейсів програмного забезпечення, орієнтовану на максимальну психологічну і зорову зручність для користувача [8].

Офіційно ISO 9241-11 визначає usability як «Ступінь з якою продукт може бути використано визначеними користувачами для досягнення визначених цілей з ефективністю, продуктивністю та вдовolenістю для специфічного контексту використання» [9].

Також юзабіліті можна описати наступним коротким терміном. Юзабіліті – це якісний показник простоти і зручності у користуванні сайтом [10].

Також не менш важливим для веб дизайну є «досвід користування». Досвід користування (англ. User Experience, UX) — це те, що людина відчуває при користуванні продуктом, системою чи сервісом (послугою). Основними об'єктами дослідження є враження, емоції та користь, отримані від взаємодії з продуктом. Також досвід користування включає практичність, простоту використання та швидкодію системи. Досвід користування має суб'єктивний

характер (бо пов'язаний з індивідуальними відчуттями і думкою) і може змінюватись з часом при зміні обставин [11].

Згідно зі стандартом ISO 9241-210, Досвід користування — це «відчуття і реакції людини, що виникли внаслідок використання або уявного використання продукту, системи чи сервісу» [9]

Юзабіліті – це функціональність, тоді як UX – це враження. Важливим є розуміти що ці два поняття дуже тісно пов'язані. І в більшості випадків погані рішення юзабіліті призводять до поганого UX [12].

У веб юзабіліті є важливою складовою для виживання. Люди залишають веб-сайт якщо він складний у користуванні. Якщо головна сторінка сайту зрозуміло не відображає чим займається компанія чи для чого призначений сайт, то люди покидають такий сайт. Якщо користувачі губляться на сайті вони покидають його. Крім того якщо інформація сайту важка для прочитання чи не відповідає їхнім цілям користувачів то вони також покидають сайт. У веб середовищі немає такого щоб користувач читав інструкції з користування сайтом чи витрачав час на те щоб якимось іншим шляхом розібратися в інтерфейсі сайту. Зараз є величезна кількість доступних сайтів, і коли користувачі зустрічаються із складністю у користуванні ресурсом то першою їхньою лінією захисту є залишення такого ресурсу [13].

За останні декілька років люди вже звикли до конкретних стандартів у веб-дизайні. І для того щоб справити серйозне враження на відвідувачів вашого ресурсу вам потрібно побудувати досвід який виходить за межі простого і зручного веб-сайту. Однак це анітрохи не означає що юзабіліті стала менш. Вона просто здобула дещо іншу роль у веб-дизайні, а саме формування основи для забезпечення найвищого рівня зручності у роботі користувача з сайтом.

Юзабіліті розуміє під собою дизайн орієнтований на користувача. І дизайн і процес розробки сфокусований навколо потенційних користувачів, тому постає завдання щоб переконатися в їхніх цілях, ментальних моделях і відповідності вимогам, і щоб створити продукт який ефективний і простий у використанні. Відповідно для забезпечення вищеописаного потрібно керуватися

певними принципами. Так можна виділити наступні 5 ключових принципів дотримання яких допоможе забезпечити хороше юзабіліті веб-сайту.

- доступність і зрозумілість;
- ясність;
- легкість в освоєні;
- надійність ;
- доречність [14].

Центральним аспектом юзабіліті є доступність вашого сайту. Якщо люди намагаються отримати доступ до вашого сайту а він неважливо з якої причини не працює то такий сайт стає безкорисним і нікому не потрібним. Такі випадки можуть призвести не лише до розчарування у користувачів але й і до втрати нових потенційних клієнтів. Можна виділити декілька основ доступності:

- час безвідмовної роботи сервера – дуже важливо переконатися що користувачі не отримують помилки при завантаженні вашого сайту, тому потрібно підібрати надійний веб-хост який забезпечить безперервну і безпомилкову роботу вашого ресурсу;
- непрацюючі посилання – також важливо переконатися що на вашому сайті працюють усі існуючі посилання;
- підтримка мобільних пристроїв – потрібно переконатися що сайт може коректно працювати на екранах з різним розширенням, а також що сайт добре працюватиме при повільному інтернет з'єднанні.

Зараз фактор підтримки сайтом мобільних пристроїв є надзвичайно важливим. Так відповідно до статистики у 2018 52,2% трафіку веб-сайтів по всьому світу зайняли мобільні телефони, крім того цей показник значно зріс у відповідності до попереднього року де він складав 50,3% [15].

Частка інших пристроїв у трафіку веб-сайтів за 2018 рік представлена на рисунку 1 [16].

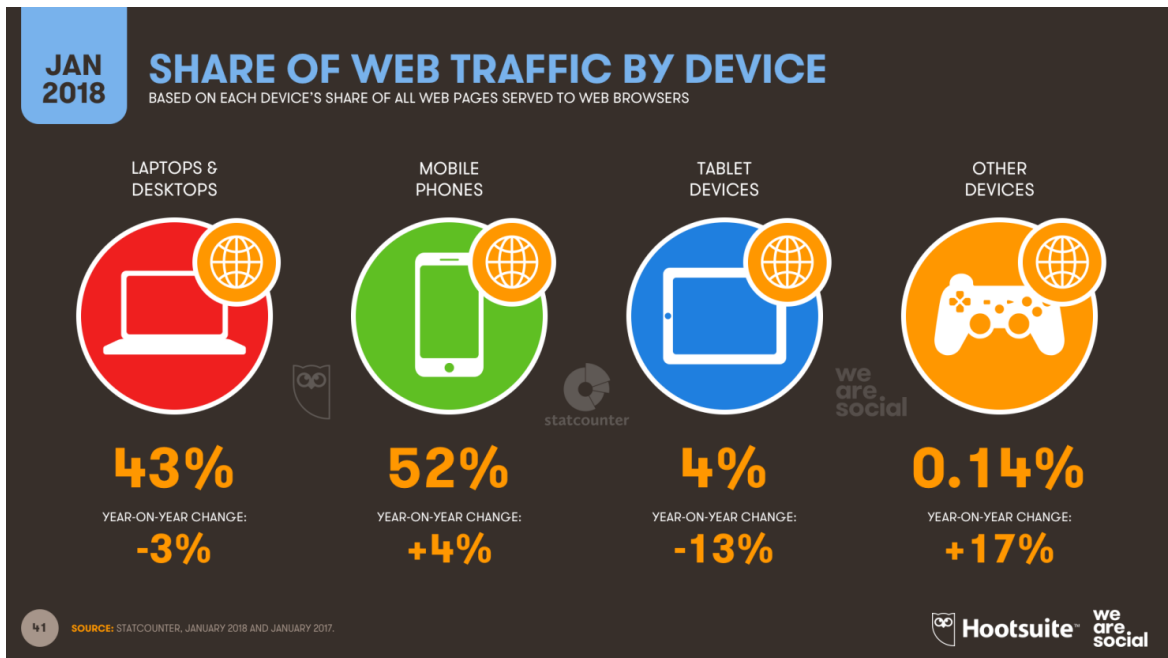


Рисунок 1.1 – Розподіл трафіку веб-сайтів по виду пристроїв за 2018 рік

Якщо ж порівняти даний показник з відповідним показником попередніх то можна прослідкувати значне зростання частки мобільних телефонів у трафіку веб сайтів. Так 2013 мобільні телефони склали 16,2% світового веб трафіку, а в 2018 цей показник зріс до 52,2% [17].

Важливо розуміти що це лише показники що представляють до уваги мобільні телефони, не беручи брати до уваги інші мобільні пристрої такі як планшети, трафік яких також збільшився за той самий період. У 2017 році загальний мобільний веб трафік складав 67% що показує що планшети значно збільшують загальне число доступу до інтернету через мобільні пристрої. [17].

Беручи до уваги вище перераховані дані очевидно що на даний момент є важливим розробити дизайн сайту який коректно працюватиме на будь якому розширені екрану сучасних пристроїв.

Принципом же який можна назвати ядром юзабіліті є ясність. Якщо ви відволікаєте чи збентежуєте ваших відвідувачів то вони або потребуватимуть більше часу щоб знайти те що вони шукають, або вони взагалі забудуть їхню початкову ціль. В будь-якому випадку вони не сприймуть ваш сайт як зручний для користувачів і скоріш за все вони підуть незадоволеними і без наміру

повернутися знову. Відвідувачі приходять на конкретний сайт з конкретними цілями на думці. І робота розробників сайту полягає в тому щоб допомогти їм досягти цих цілей настільки швидко наскільки лише можливо. Якщо ви можете це забезпечити то ваші відвідувачі будуть задоволені і ви закладете фундамент для позитивного досвіду в користувача від користування вашим сайтом. [14]

Серед головних складових які можуть допомогти створити зручний і зрозумілий сайт можна виділити наступне:

- простота – це означає що потрібно сфокусуватися на тому що є важливим. якщо ви не відволікатимете ваших користувачів то вони з більшою ймовірністю використовуватимуть сайт за призначенням яке ви в нього заклали;

- знайомість – при розробці дизайну потрібно притримуватися певних загальноприйнятих шаблонів розміщення елементів сайту з якими користувачі вже добре знайомі. В таких випадках немає нічого неправильного щоб шукати натхнення в дизайні інших сайтів;

- послідовність – потрібно створити послідовний досвід по усьому вашому сайту щоб користувачі відчували себе невимушено під час користування ним;

- безпосередній зворотний зв'язок – зворотній зв'язок є необхідним для будь-якої взаємодії. Коли люди взаємодіють з вашим сайтом, вам потрібно забезпечити можливість користувачу легко залишити відгук про те чи було його користування сайтом успішним чи ні;

- хороша архітектура інформації – важливо розуміти ментальні моделі ваших користувачів і в якому вигляді вони очікують від вас що ви структуруєте контент на вашому сайті.

Легкість в освоєні є іншим не менш важливим аспектом юзабіліті. Ціллю розробників повинна бути розробка інтуїтивного інтерфейсу – інтерфейсу який не вимагає інструкцій, чи то навіть довгого процесу спроб і помилок щоб розібратися в ньому. Ключ до інтуїтивного дизайну це використовувати те що люди вже знають або створити щось нове що буде легке в освоєні. Зараз люди знайомі з багатьма концепціями дизайну які використовуються у веб. Тож

використовуючи ці концепції послідовно ви відповідаєте очікуванням ваших користувачів. Таким чином ви допомагаєте їм досягнути своїх цілей набагато швидше. Людська природа включає те що ми любимо шаблони і образи, саме тому ми краще справляємося зі знайомими ситуаціями краще ніж з незнайомими. Тож при використанні нових концепцій у дизайні сайту необхідно переконатися в послідовному їх використанні і крім того потрібно забезпечити людям допомогу на початковому етапі їх освоєння. Для цього можна запропонувати додаткову інформацію чи інструкції на випадок коли користувачі користуються ресурсом вперше. Також не менш важливо щоб нова концепція дизайну була простою і наглядною щоб як можна легше запам'ятатися користувачу.

У веб дизайні є конкретні умовності з якими протягом років інтернет користувачі стають все більше і більше знайомі. Такі умовності включають:

- наявність головної навігації яка розміщена у верхній, або в рідких випадках в лівій частині сторінки;
- наявність логотипу у верхньому лівому куті або центрі сторінки
- наявність логотипу на який можна натиснути і таким чином перейти на головну сторінку сайту;
- наявність посилань які змінюють колір або зовнішній вигляд коли ви наводите на них курсор [14].

Ключовим принципом юзабіліті будь-якого сайту є надійність. За статистикою 75% судження про надійність бізнесу у користувачів формується базуючись на дизайні їхнього веб-сайту [18].

Навіть якщо люди знаходять контент який вони шукають, якщо вони не довіряють вам то від цього контенту немає ніякої користі. Ваш сайт може змушувати ваших відвідувачів скептично ставитися до вашого бізнесу у різний спосіб включаючи сумнів щодо вашого існування, репутації чи якості вашого контенту. Важливо щоб люди знали що сайт представляє реальна організація з реальними людьми. Тому важливо додати на сторінку «About us» наряду з контактами і фізичною адресою. Звичайно контент грає важливу роль для сприйняття надійності вашого сайту. Зважаючи на це необхідно переконатися у

чесності і точності вашого контенту. Потрібно уникати помилок таких як неправильна граматики чи друкарські помилки.

Не менш важливим принципом який необхідний для хорошої юзабіліті сайту є доречність. Недостатньо щоб сайт був зрозумілим, важливо також щоб ваш контент був доречним. Знову ж таки необхідно щоб ви знали ваших користувачів і чому вони відвідують ваш сайт. Тому при розробці дизайну сайту потрібно почати з визначення ким є ваші потенційні користувачі. Крім того потрібно з'ясувати якими є їхні цілі при відвідуванні вашого сайту. Також потрібно визначити сценарії користувачів які демонструють в яких ситуаціях люди відвідують ваш сайт щоб знайти певний контент. Тому яке б дизайнерське рішення не виконувалося завжди потрібно орієнтуватися на те щоб в результаті ви отримали сайт який зручний в використанні для користувача.

По суті поняття юзабіліті та UX дещо перетинаються тому аспекти які мають значення для забезпечення юзабіліті сайту також в певній мірі стосуються UX. Однак поняття UX є дещо ширшим і тому містить інші принципи такі як:

- простота;
- візуальна ієрархія;
- навігація;
- орієнтація на користувача.

Простота важливий аспект UX коли говорити про веб-сайт. Хоча зовнішній вигляд вашого сайту і важливий однак більшість відвідувачів не приходять до нього щоб подивитися наскільки в нього блискучий дизайн. Замість того вони відвідують ваш сайт щоб здійснити якусь дію, або щоб знайти певну інформацію. Додавання непотрібних елементів дизайну(елементів які не мають ніякої функціональної цілі) до вашого сайту робить важчим для користувача виконання тих дій які він мав виконати на вашому сайті.

Простоту сайту можна забезпечити у ряд різних способів ось декілька прикладів:

– не використовувати занадто багато кольорів. Згідно з книгою *The Handbook of Computer-Human Interaction* рекомендується максимум 5 різних кольорів(плюс чи мінус два) у дизайні веб-сайту [19];

– шрифти які ви використовуєте мають бути як мінімум розбірливими. Загальні рекомендації щодо шрифтів говорять про те що варто використовувати максимум три різних гарнітури шрифтів, а також максимум трьох різних розмірів [20];

– графіка. Її потрібно використовувати лише якщо вона допомагає користувачу виконати якесь завдання чи коли виконує якусь конкретну функцію. Не потрібно просто за бажанням додавати графіку в дизайн де попало.

Окрім того від простоти сайту залежить те наскільки швидко він буде завантажуватися а це важливо для приваблення нових користувачів і не тільки. Так, за статистикою показник відмов веб-сайту різко зростає до 100% якщо завантаження сторінки займає 4 секунди і більше [21].

Як правило особливості які сповільнюють роботу сайту це складні зображення, функції та параметри. Усунувши їх, мінімалістичний сайт буде завантажуватися на неймовірно високій швидкості і буде приваблювати нових користувачів і більше трафіку в пошукових системах [22].

Існує декілька висновків про важливість наявності простоти в дизайні сайту які підкріплені солідними дослідженнями. Так серед найбільш яскраво виражених висновків є те що простому дизайну надають перевагу над складним. Дослідження проведене в Гардвардському університеті показало що колір і складність сайту складають 48% рейтингу візуальної привабливості сайту [23].

Візуальна ієрархія складова UX важливість якої зросла особливо в останнє десятиліття. Близько пов'язана з принципом простоти, візуальна ієрархія розуміє під собою розташування і організацію елементів сайту таким чином щоб в першу чергу відвідувачі природно тягнулися до найбільш важливих елементів. Коли справа доходить до оптимізації сайту у відповідності до юзабіліті і UX, цілю є те щоб спонукати відвідувачів здійснити бажану дію,

однак це має відбуватися природно і невимушено. Коректуючи розташування, розмір і колір певних елементів можна структурувати сайт таким чином щоб увага відвідувачів була привернута до цих елементів в першу чергу.

Навігація. Наявність інтуїтивної навігації на вашому сайті є ключовим для того щоб гарантувати що відвідувачі зможуть знайти що вони шукають. В ідеалі користувач має бути здатен вперше прийшовши на сайт сильно не роздумуючи де далі він має натиснути знайти те що він шукає, тобто перехід між сторінками сайту має бути максимально легким.

Загалом можна виділити декілька шляхів як оптимізувати навігацію сайту серед них такі:

- зберігати простоту структури основної навігації сайту і розташовувати її вгорі сторінки;
- включити навігацію у футер(footer) сайту;
- використовувати навігаційні ланцюги(breadcrumbs) на кожній сторінці окрім головної для того щоб користувач був обізнаний про те в якому місці сайту він знаходиться;
- розташувати поле пошуку близько до верхньої частини вашого сайту щоб користувач легко міг здійснити пошук;
- не пропонувати занадто багато навігаційних опцій на сторінці;
- не робити навігацію занадто глибокою. Найкращим є робити навігацію вашого сайту не більше ніж три рівневою;
- також важливо щоб головна навігація вашого сайту на протязі усіх сторінок сайту залишалася однаковою [24].

Одна з найбільш важливих частин сайту це навігаційне меню. Панелі меню навігації зазвичай містять інформацію таку яка ваші послуги, контакти, про вас, блог і так далі. Але хороша навігація має бути супер легкою і простою щоб зайти і зрозуміти користувачу потрібну інформацію на вашому сайті Меню навігації може буквально покращити чи зруйнувати ваш сайт, тож дуже важливо щоб воно функціонувало належним чином, залишаючись простим і не перетворювало ваш сайт в заплутаний лабіринт при користуванні користувачами навігацією [23].

Послідовність складова яка є важливою в UX. Так на додачу до збереження послідовності в навігації сайту, загальний вигляд і враження вашого сайту мають бути послідовними на протязі всіх сторінок вашого сайту. Фон, схеми кольорів, шрифти і навіть тон вашого письма і загалом всі області де послідовність може здійснити позитивний вплив на юзабіліті та UX. Однак це не означає що кожна сторінка вашого сайту повинна мати абсолютно однаковий макет. Натомість вам варто створити різні макети для конкретних типів сторінок(наприклад макет для цільових сторінок і макет для інформаційних сторінок) і використовуючи ці макети послідовно, таким чином ви дозволяєте користувачам легше розуміти який тип інформації вони скоріш за все знайдуть на представленій сторінці.

Наступна складова UX яка вже так чи інакше згадувалася це орієнтованість на користувачів. Тому що якщо ви не розробляєте сайт для користувача то для кого ви тоді його розробляєте. Тому хоча вище описані принципи є важливими для покращення юзабіліті і UX, однак реальним ключем до покращення дизайну сайту є проведення тестування сайту користувачами, збір відгуків і внесення змін на основі того що ви дізналися.

1.3.2. Аналіз сучасних підходів до реалізації системи

Важливою складовою юзабіліті та UX сучасних веб сайтів який в певній мірі є окремою ланкою дизайну сайту є адаптивний веб-дизайн. Термін адаптивний веб дизайн дає ідею щодо розробки і проектування сайтів які працюватимуть на будь якому пристрої змінюючи свій макет автоматично.

З винаходом все більшої і більшої кількості нових технологій і нових пристроїв користувачі хочуть використовувати найкращі і найновіші пристрої. У минулому були лише настільні комп'ютери які кожен використовував. Тож не було важливим для дизайнерів працювати над адаптивним веб дизайном. Однак у сучасному світі існує багато різноманітних пристроїв таких як мобільні телефони, планшетні комп'ютери і розумні телевізори інші пристрої.

Зараз реальний факт полягає в тому що в сучасному світі люди хочуть переглядати веб сторінки легко і швидко. Це означає що адаптивний веб дизайн

був створений щоб допомогти людям переглядати сайти з будь якого типу пристрою який вони мають у своїх руках. Веб дизайнери мають розробляти такі веб-сайти які представляють інформацію на екранах різних розмірів і на екранах з різним розширенням користувачам. Ваш сайти має бути багатостороннім щоб бути здатним використовуватися і на мобільному телефоні і на величезному екрані. До створення адаптивного веб дизайну розробникам сайтів потрібно було створювати багато версій їхнього сайту що було дійсно складним однак це було єдиним рішенням. Саме тому в наш час має величезну популярність і важливість оскільки дозволяє відповідати всім цим вимогам і при цьому не потребує важкої роботи щодо розробки багатьох версій одного й того самого сайту однак для різних пристроїв [25].

Ідея на якій базується адаптивний дизайн полягає в тому щоб використовувати один і той самий HTML код незважаючи на те який пристрій використовувався для того щоб отримувати доступ до сторінки. В такому випадку потреба створювати окремі веб-сайти відпадає. При цьому одні й ті ж самі елементи сайту в залежності від розміру екрану розташовуються певним чином [26].

Зараз адаптивні веб сторінки стали все більше стандартною практикою аніж минаючою тенденцією і все це розпочалося з того що дизайнери в ранніх 2000-х гралися над ідеєю пристосування макету їх веб сторінок у відповідності до ширини браузера [27].

Однак початок насправді широкого розвитку і застосування адаптивний дизайн набув після виходу у 2010 році коли веб-дизайнер Ітан Маркотт започаткував термін «адаптивний дизайн(responsive design)» в статті для A List Apart [28].

По суті в статті пропонується звернутися до ландшафту пристроїв, браузерів, розмірів екранів та орієнтацій які постійно змінюються шляхом створення гнучких, плавних і адаптивних веб-сайтів. Замість того щоб задовольняти тогочасні потреби у версіях для настільних комп'ютерів які адаптовані під найбільш використанні розширення екранів, наряду з мобільною версією(часто під певний єдиний мобільний пристрій) ідея полягала в тому щоб

підійти до проблеми у інший спосіб. А саме використовувати гнучкі та плавні макети які адаптуються практично під будь-яке розширення екрану.

Відповідно до цього в основі адаптивного веб дизайну лежать три наступні технічні функції:

- медіа черги і слухачі медіа черги;
- гнучкий макет який базується на сітці і використовує відносний розмір;
- гнучкі зображення і медіа які створюються з допомогою динамічного змінення розміру чи CSS [29].

1.3.3. Аналіз сучасних засобів реалізації системи

Однак однією справою є створити дизайн сайту а зовсім інша справа його програмно реалізувати. Зараз створення сайту з нуля використовуючи лише чистий HTML, CSS та JavaScript код є досить затратним не лише з точки зору часу але й з точки зору ресурсів, особливо якщо мова йде про реалізацію адаптивного веб дизайну. Тому зараз великої популярності набирає тенденція створення сайтів на основі фреймворків.

Фреймворк (англ. Framework) – це стандартизований набір концепцій, застосувань і критеріїв для вирішення загального типу проблем, який може використовуватися в якості стандарту щоб допомогти нам підходити і вирішувати нові проблеми аналогічного характеру [30].

У світі веб дизайну щоб дати більш просте визначення фреймворк визначають як пакет який складається із структури папок файлів і папок стандартизованого коду (HTML, CSS, JS документи і т.д.) які можуть бути використані в якості основи для початку розробки сайту [31].

CSS фреймворки забезпечують базову структуру для розробки послідовних рішень для боротьби з проблемами які постійно зустрічаються в front-end розробці сайтів. Вони забезпечують загальну функціональність яка може бути перевизначена для конкретних сценаріїв і додатків. Це значним чином зменшує час який потрібно витратити на початок створення додатків і веб-сайтів. [30]

Таким чином розробникам не потрібно кожного разу з нуля починати розробку сайтів. Вони можуть повторно використати базовий фундамент від попередніх сайтів і відразу приступити до роботи над головними складовими веб-сайту аніж знову прописувати код для кожної дрібниці знову і знову для кожного наступного сайту. [30]

В певній мірі фреймворк подібний у своєму призначенні до бібліотеки однак між ними є певні відмінності. Ключова відмінність між бібліотекою і фреймворком це те де розташований елемент управління. В бібліотеці контроль покладається на того хто її викликає, тобто на вас. Це означає що ви контролюєте те коли бібліотеці потрібно виконувати конкретну функцію а решта коду окрім цієї задачі є незалежною від бібліотеки.

У випадку ж фреймворку управління здійснюється навпаки. Потік управління знаходиться всередині фреймворку і ви можете налаштувати його у місцях які підходять вашому випадку використання. [30]

Дана відмінність у структурі фреймворку і бібліотеки представлено на рисунку 1.2.



Рисунок 1.2 – Структура фреймворку і бібліотеки відповідно [32]

Заздалегідь визначена структура яку фреймворки забезпечують допомагає підтримувати послідовність на протязі всього сайту і різноманітних його елементів. Пакет який надається у front-end фреймворках зазвичай містить HTML, CSS і JavaScript файли для того щоб гарантувати що різні сторінки протягом всього сайту мають однаковий дизайн, типографію, форми і т.д.

Коротко про переваги фреймворку:

- більш простіша підтримка коду;
- узгоджена організаційна структура для проекту;
- адаптивні медіа-запити;
- одноманітний стиль серед підказок до інструментів, кнопок, форм

та інших елементів. [30]

Іншим фактором є те що фреймворки дозволяють забезпечити розробникам певну кількість незалежності під час роботи над проектами. Розробникам не обов'язково отримувати специфікації від дизайнерів для кожного окремого елемента. Вони мають послідовну схему дизайну яка забезпечується вродженою структурою і дозволяють виконати налаштування для конкретних випадків, як тільки це є необхідним. Однак це не означає що дизайнери стали зайвими після появи фреймворків. Це просто означає що розробникам не потрібно консультуватися щодо правок чи вказівок по дизайну кожного елемента щоб досягнути однакового стилю з точністю до пікселя.

Дизайнери ж можуть скоротити час який вони витрачають на розробку дизайну для базових елементів таких як кнопки чи форми і натомість сконцентруватися на більш пріоритетних речах в дизайні. Таким чином фреймворки дозволяють зменшити додаткові витрати на взаємодію дизайнера і розробника. [30]

Більшість веб-сайтів розділяють дуже просту якщо не сказати ідентичну структуру. Ціль фреймворків полягає в забезпеченні загальної структури для того щоб розробникам не потрібно було переробляти все з нуля і щоб можна було знову використовувати наданий код. Таким чином фреймворки дозволяють нам скоротити великі обсяги роботи і таким чином зберегти багато часу.

В принципі є два види фреймворків такі як backend і frontend фреймворки. Важливим є розуміти що фреймворк це концептуальне поняття: заздалегідь підготовлений набір з якого починають працювати. Концепція фреймворку може бути застосована до різних процесів які виконуються у веб наприклад таких як: програмістський рівень який з'єднує базу даних з

контентом сайту і використовує мову PHP, і дизайнерський рівень, де контент має бути презентований в HTML документах з визначеними таблицями стилів CSS щоб в кінцевому рахунку бути переглянутим в браузері. [31]

Front-end фреймворки або CSS фреймворки зазвичай складаються з пакету що складається із структури файлів і папок стандартизованого коду. Звичайно фреймворки містять наступні елементи:

- вихідний код CSS для створення сітки: це дозволяє розробникам позиціонувати різні елементи які роблять дизайн сайту простим і універсальній манері;
- визначення стилів типографії для HTML елементів;
- рішення для випадків несумісності щоб сайт відображався коректно в усіх браузерах;
- створення стандартних класів CSS які можуть бути використані для того щоб стилізувати додаткові компоненти інтерфейсу користувача [31].

Сучасний розвиток методів адаптивного веб дизайну, які полегшують розробку сайтів які можуть адаптуватися до різних розширень різних мобільних і настільних пристроїв приводить до появи адаптивних фреймворків.

Серед CSS фреймворків ми можемо провести відмінність між двома типами фреймворків зважаючи на їхню складність, а саме є прості і повноцінні фреймворки. Ця відмінність є суб'єктивною і не означає що один тип фреймворку кращий швидше можна сказати що вони дають рішення різного рівня в залежності від того який рівень складності і гнучкості необхідний.

Прості фреймворки часто називають «системи сітки» (англ. grid systems) однак тим не менше вони є фреймворками. Вони пропонують таблиці стилів з системами стовбців щоб полегшити розподіл різних елементів відповідно до створеного дизайну.

Повноцінні фреймворки зазвичай пропонують функції які можна налаштувати такі як стилізована типографія, набір форм, кнопок, іконок і багаторазові компоненти побудовані для того щоб забезпечити навігацію, оповіщення, HTML шаблони, користувацькі налаштування і багато іншого.

Розглянувши фреймворки детальніше можна виділити те які у них є переваги і які є недоліки.

Переваги використання фреймворку:

- пришвидшення процесу реалізації сайту;
- чистий і акуратний код;
- наявність рішень для загальних проблем CSS;
- сумісність з браузерами;
- наявність єдиної процедури для вирішення загальних проблем робить ведення різних проектів більш простішим;
- значна користь в командній роботі над проектом;

Недоліки використання фреймворку:

- змішування контент і представлення;
- наявність залишкового коду що не використовується;
- більш повільніше вивчення;
- без ресурсів для вивчення фреймворку важко його освоїти.

Чи обов'язково використовувати фреймворк? Насправді ні, фреймворки це ресурси що можуть бути надзвичайно корисними для багатьох людей однак це не означає що вони обов'язково будуть корисними для конкретного проекту. Однак наразі можна рідко зустріти проект який не потребує використання фреймворку тому що у більшості випадків існує багато причин щоб створювати веб-ресурс саме на базі фреймворку. Однак в залежності від проекту який реалізовується потрібно обрати найбільш підходящий для найкращого вирішення завдання фреймворк.

Які при виборі будь яких речей для якогось конкретного застосування, так і при виборі фреймворку потрібно керуватися певними критеріями і вимогами у відповідності до того що планується створювати з допомогою фреймворка. Крім того правильний front-end фреймворк спрощує і пришвидшує процес проектування і розробки веб-сайту і в той же час надає вам гнучкість та можливості необхідні для досягнення виняткового результату. Розуміючи те які властивості повинен мати фреймворк для вирішення вашої задачі буде набагато

простіше вибрати серед великого їхнього різноманіття зважуючи їхні переваги і недоліки. При виборі фреймворку можна керуватися наступними критеріями:

- рівень навичок – важливо брати до уваги ваші навички вибираючи front-end фреймворк. Якщо ви новачок то вам може підійти більш міцний фреймворк такий як Bootstrap, він являє собою готовий набір багатьох корисних функцій і при цьому вимагає мінімальні навички написання коду. Якщо ви більш досвідчений вам може краще підійти простіший фреймворк який дає вам широкі можливості з точки зору налаштувань під конкретне застосування. Такі фреймворки зазвичай компактніші і менш громіздкі що в даному випадку є плюсом;

- адаптивний дизайн – будь який сайт який ви розробляєте повинен відображатися належним чином на всіх пристроях тому що все більше і більше людей отримують доступ до мережі інтернет через мобільні пристрої. Відповідно до даних за 2018 рік 52,2% трафіку користувачів на сайтах приходиться на мобільні пристрої. [15] Тому варто використовувати front-end фреймворки які підтримують адаптивний веб дизайн щоб таким чином мати менше клопоту з адаптуванням сайту до відображення на різних пристроях;

- CSS препроцесори – якщо ви використовуєте CSS препроцесори і надаєте перевагу тому чи іншому препроцесору, зазвичай це SASS чи LESS то варто переконатися що фреймворк його підтримує;

- зовнішній вигляд – варто обирати фреймворк що дозволяє вам досягнути такого зовнішнього вигляду який ви хочете докладаючи при цьому мінімальних зусиль;

- прототипи – ідеальний front-end фреймворк дозволяє вам швидко створювати каркаси і прототипи для пришвидшення загального процесу проектування [33];

- найкраща довгострокова підтримка – багато цифрових проектів в тому числі не завжди довго існують, як наслідок оновлення і підтримка сервісів через деякий час зупиняється. Тож завжди краще робити вибір в користь фреймворку який гарантує тривалу підтримку. Багато з них підтримуються компаніями що пропонують інші професійні продукти на ринку.

Розгляньмо одні з найкращих сучасних фреймворків з допомогою яких можна реалізувати систему.

1) Огляд фреймворку Bootstrap

Приємний, інтуїтивний і потужний front-end фреймворк для швидшої та простішої веб розробки. Фреймворк Bootstrap який початково називався Twitter Blueprint був розроблений Марком Отто та Якобом Торнтоном в компанії Twitter і вперше був випущений у 2011 році [34].

В таблиці 1.1 представлено загальну інформацію про фреймворк Bootstrap.

Таблиця 1.1 – Загальна інформація про фреймворк Bootstrap

Творці	Марк Отто і Якоб Торнтон
Рік появи	2011
Популярність	126,202 зірок на GitHub
Препроцесор	Sass
Адаптивний	Так
Початкові шаблони/макети	Так
Документація	Наявність найбільшої кількості ресурсів для вивчення серед усіх фреймворків
Підтримка браузерів:	Останні версії Firefox, Chrome, Safari, IE810-11-Microsoft Edge.

На рисунку 1.3 представлено веб-сторінку фреймворку.

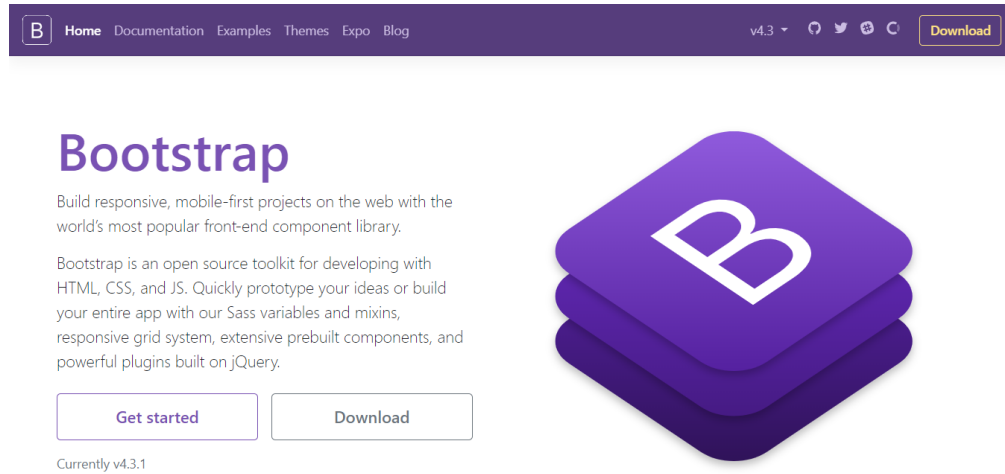


Рисунок 1.3 – Веб-сторінка фреймвоку Bootstrap [35]

Розроблений в компанії Twitter, Bootstrap використовує LESS CSS, компілюється через Node, і управляється через GitHub щоб допомогти ентузіастам робити надзвичайні речі у Веб [36]. Початково він був розроблений як інструмент для внутрішніх команд, однак як тільки його було випущено його застосування значно розширилось. Він надає шаблони дизайну для таких загальних компонентів UI як кнопок, типографії, форм, випадаючих меню, сповіщень, вкладок, слайдів а також додаткові розширення JavaScript.

Зараз це фреймворк з відкритим кодом який найбільше використовується по усьому світі. Bootstrap притримується стандартів адаптивного веб дизайну тому дозволяє розробляти адаптивні веб-сайти будь якої складності і розмірів. [33]

Завдяки тому що Bootstrap постійно оновлюється то він зазвичай включає найновіші та найкращі функції. Так наприклад до нього були додані теми які відповідають принципам Google's material design незадовго після того як вони тільки були опубліковані, а також він був оновлений для того щоб можна було використовувати Sass як CSS препроцесор. [33]

Враховуючи його величезну популярність яка досі зростає можна бути впевненим що цей чудовий інструментарій не підведе тебе і не залишить тебе наодинці на твоєму шляху по створенню успішного веб-сайту [37].

Головна сила Bootstrap це його величезна популярність. Технічно він не обов'язково кращий ніж фреймворки які перераховані нижче, однак він пропонує набагато більше ресурсів(статті та навчальні матеріали, сторонні плагіни і розширення, засоби для створення тем і багато іншого) ніж інші front-end описані нижче разом взяті. Іншими словами Bootstrap всюди і це головна причина чому люди продовжують обирати його. [37]

Даний фреймворк має наступні переваги та недоліки.

Переваги:

- послідовний фреймворк який підтримує основні з усіх браузерів і CSS виправлення сумісності ;
- підтримка адаптивного веб дизайну(може бути відключена якщо є така необхідність);
- велика детальна документація та підтримка спільноти;
- чудова система сітки [38].

Недоліки:

- нестандартний розмір файлу у 276 КБ через надмірну кількість стилів що рідко використовуються [33];
- надмірна кількість HTML класів та елементів DOM можуть бути заплутаними і викликати безлад в коді;
- якщо потрібно створити відмінний від стилів Bootstrap сайт перевизначення стилів може привести до великих затрат часу.

Зважаючи на сильні і слабкі сторони Bootstrap буде ідеальним для початківців і тих хто надає перевагу надійним front-end фреймворкам. [33]

2) Огляд фреймворку Foundation 3

Створений компанією з веб дизайну Zurb, Foundation є високо передовим front-end фреймворком корпоративного класу який ідеальний для розробки швидких адаптивних веб-сайтів. При тому що він використовується на таких сайтах як Facebook, eBay і Mozilla цей фреймворк є доволі складним і може не підійти для новачків. [33]

В таблиці 1.2 представлено загальну інформацію про фреймворк Foundation.

Таблиця 1.2 – Загальна інформація про фреймворк Foundation

Творці	ZURB
Рік появи	2011
Популярність	28.4 зірок на GitHub
Препроцесор	Sass
Адаптивний	Так
Початкові шаблони/макети	Так
Документація	Хороша, з доступними багатьма додатковими ресурсами
Підтримка браузерів:	Firefox, Chrome, Safari, IE9+

На рисунку 1.4 представлено веб-сторінку фреймворку.

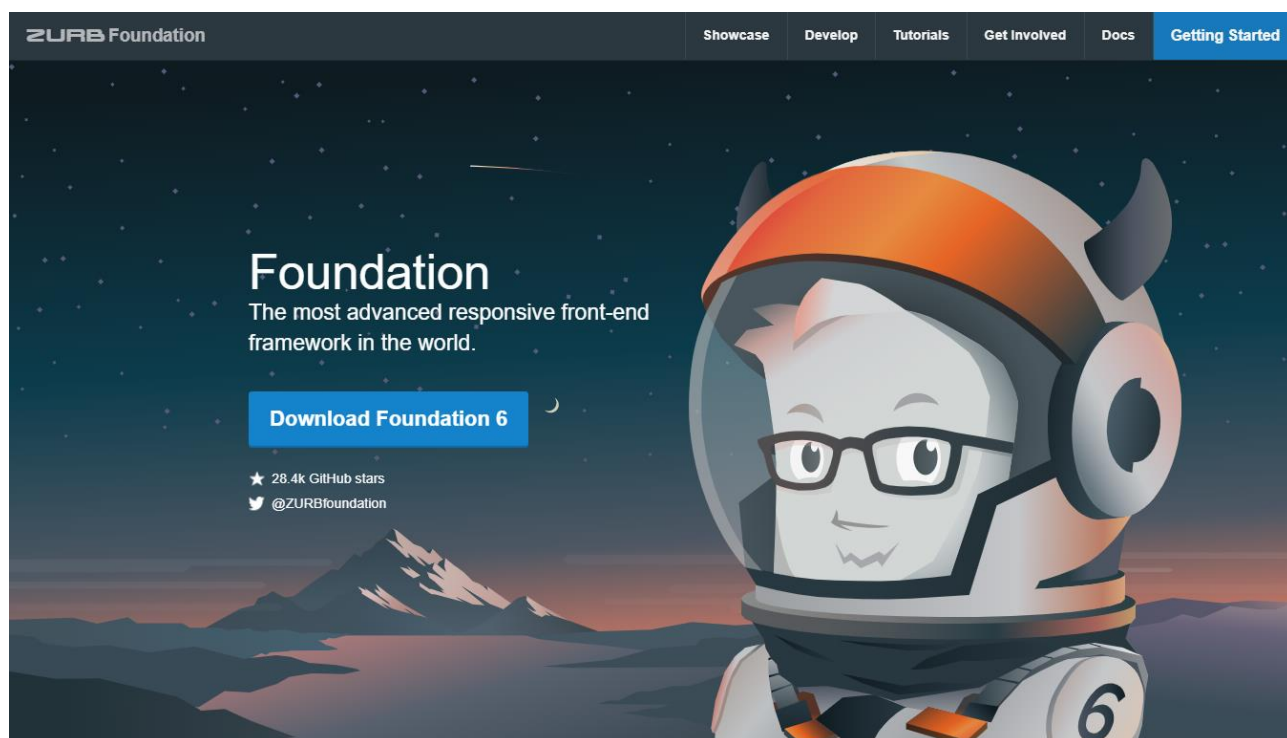


Рисунок 1.4 – Веб-сторінка фреймвоку Foundation [39]

Цей функціонально багатий фреймворк підтримує GPU прискорення для плавної, блискавичної анімації і Fastclick.js для швидкого відображення на мобільних пристроях. Foundation побудований на Sass, потужному CSS процесорі і включає розроблений ним атрибут обміну даними який дозволяє завантажувати легші секції HTML на мобільних пристроях і важчі секції HTML на пристроях з великими екранами. [33]

Foundation дійсно професійний front-end фреймворк з підтримкою бізнесу, навчанням і наданням консультацій. Він також надає багато ресурсів щоб допомогти новачкам вивчити і використовувати фреймворк швидше і простіше.

Серед переваг та недоліків Foundation можна виділити наступні.

Переваги:

- відсутність блокування стилю, тому є більша гнучкість;
- використовує REM замість пікселів усуваючи необхідність явно задавати ширину, висоту і інші атрибути для кожного пристрою;

Недоліки:

- доволі великий розмір файл фреймворку;
- дещо складний для початківців.

Даний фреймворк добре підходить для розробників які мають пристойний досвід і які початково сконцентровані на розробці швидких, привабливих і адаптивних веб-сайтів. [33]

3) Огляд фреймворку Semantic-UI

Будучи відносним новачком Semantic-UI виділяється декількома аспектами і може стати одним з найбільш популярних front-end фреймворків. Semantic UI це постійні зусилля зробити створення сайтів на багато більше семантичним. Він використовує принципи природної мови таким чином роблячи код на багато більше читабельним та зрозумілим.

В таблиці 1.3 представлено загальну інформацію про фреймворк Bootstrap. [33]

Таблиця 1.3 – Загальна інформація про фреймворк Semantic UI

Творці	Jack Lukic
Рік появи	2013
Популярність	46,859 зірок на GitHub
Препроцесор	Less
Адаптивний	Так
Початкові шаблони/макети	Так
Документація	Дуже хороша, пропонується добре структурована документація плюс є декілька сайтів які пропонують інструкції для початку роботи, налаштування і створення тем.
Підтримка браузерів:	Firefox, Chrome, Safari, IE10+

На рисунку 1.5 представлено веб-сторінку фреймворку Semantic UI.



Рисунок 1.5 – Веб-сторінка фреймворку Semantic UI [40]

Головною особливістю даного фреймворку є його простота. Завдяки тому що він використовує природну мову його код не потребує пояснення. Таким чином процес розробки дещо простіший і більш впорядкований. Тому навіть розробники які мають невеликий досвід в написанні коду можуть легко почати з ним працювати.

Іншою помітною функцією Semantic-UI є його інтегрованість з неймовірною кількістю сторонніх бібліотек.

Semantic UI це найбільш інноваційний і повнофункціональний front-end фреймворк серед описаних тут. Загальна структура фреймворку і узгодженість імен з точки зору чіткої логіки і семантики його класів також краща ніж в інших фреймворків. [37]

Серед переваг та недоліків фреймворку Semantic UI можна виділити наступні.

Переваги:

- семантичні імена класів формують низький вхідний бар'єр тому навіть новачки можуть почати працювати з фреймворком;
- невеликі розміри файлів і мінімальний час завантаження у зв'язку з тим що можна завантажувати лише компоненти які вам потрібні, при цьому кожен з них має власний JS файл і таблицю стилів;
- різносторонні елементи дозволяють легко проводити різні налаштування.

Недоліки:

- дуже великі пакети в порівнянні з Foundation та Bootstrap;
- для того у кого більш складніші потреби в дизайні і розробці цього фреймворку може бути недостатньо.

Ідеальний для початківців які шукають легкий і швидкий фреймворк. [33]

4) Огляд фреймворку Materialize

Materialize адаптивний фреймворк для front-end розробки який реалізує специфікації Google's material design і завантажується з готовими кнопками,

значками, картами, формами та іншими компонентами. Він пропонується як і в стандартній так і у версії яка працює на базі SASS.

В таблиці 1.3 представлено загальну інформацію про фреймворк Bootstrap.

Таблиця 1.3 – Загальна інформація про фреймворк Materialize.

Творці	Alvin Wang
Рік появи	2014
Популярність	36,827 зірок на GitHub
Препроцесор	Sass
Адаптивний	Так
Початкові шаблони/макети	Так
Документація	Хороша
Підтримка браузерів:	Firefox 31+, Chrome 35+, Safari 7+, IE 10+

На рисунку 1.5 представлено веб-сторінку фреймворку Materialize.



Materialize

A modern responsive front-end framework
based on Material Design

GET STARTED

UPGRADE FROM 0.100.2

Release: 1.0.0

Рисунок 1.5 – Веб-сторінка фреймворку Materialize [41]

Materialize включає зручну функцію сітки колонок `IZ` які можуть бути використані для макетів веб-сайтів. Він також завантажується з `CSS` який готовий для використання за замовчуванням для `material design shadows`, типографії, кольорів та інших функцій.

Додаткові функції включають анімацію з ефектом пульсації, `drag-out` меню для мобільних, домішки `SASS` і багато іншого. [33]

Серед переваг та недоліків фреймворку Materialize можна виділити наступні:

Переваги:

- великий вибір компонентів;
- підтримка адаптивності гарантує те що сайти будуть працювати на всіх пристроях.

Недоліки

- великий розмір файлу робить даний фреймворк доволі громіздким;
- відсутня підтримка для моделі `Flexbox`.

Зважаючи на його характеристики можна сказати що він добре підійде для розробників з невеликим досвідом яким потрібне керівництво для роботи із специфікаціями `Google's material design`. [33]

5) Огляд фреймворку Material UI

Якщо потрібен `front-end` фреймворк з допомогою якого легко дотримуватися принципів `Google's material design` то важко помилитися в даному випадку з `Material UI`. Це безумовно досі найретельніше розроблений фреймворк для реалізації цих принципів однак тут є одне але, цей фреймфорк не повинен бути відправною точкою для абсолютно нового проекту веб-дизайну. [33]

В таблиці 1.3 представлено загальну інформацію про фреймворк `Bootstrap`.

Таблиця 1.3 – Загальна інформація про фреймворк `Material UI`.

Творці	material-ui.com
Рік появи	2014
Популярність	52,673 зірок на GitHub
Препроцесор	Less
Адаптивний	Так
Початкові шаблони/макети	Так
Документація	Хорошая
Підтримка браузерів:	Firefox, Chrome, Safari

На рисунку 1.5 представлено веб-сторінку фреймворку Material UI.

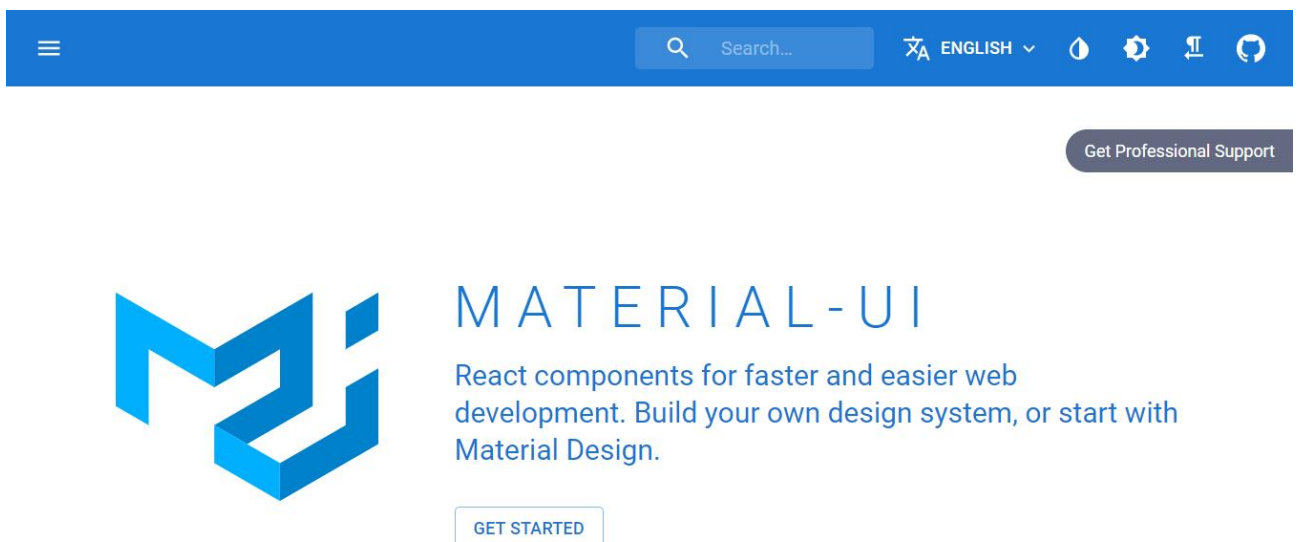


Рисунок 1.5 – Веб-сторінка фреймворку Material UI [42]

Мітючи готові CSS і сумісні з material design компоненти Material UI використовує рішення CSS-in-JS. Це дозволяє використовувати багато чудових функцій таких як вкладеність тем, динамічні стилі, самостійність і т.д. [33]

Серед переваг та недоліків фреймворку Material UI можна виділити наступні.

Переваги:

- найпростіший спосіб реалізувати принципи Google's material design використовуючи фреймворк;
- великі можливості для налаштування під конкретні задачі.

Недоліки:

- не призначений для того щоб слугувати відправною точкою для реалізації проектів веб-дизайну з нуля;
- потребує серйозне розуміння React для ефективного використання [33].

б) Огляд фреймворку UIKit

UIKit це високо модульний front-end фреймворк який вирізняється серед інших front-end фреймворків для веб розробки з багатьох причин. Головна з них це те що він включає і LESS, і SASS препроцесори.

В таблиці 1.3 представлено загальну інформацію про фреймворк Bootstrap.

Таблиця 1.3 – Загальна інформація про фреймворк UIKit.

Творці	YOOtheme
Рік появи	2013
Популярність	15,164 зірок на GitHub
Препроцесор	Less, Sass
Адаптивний	Так
Початкові шаблони/макети	Так
Документація	Хороша
Підтримка браузерів:	Chrome, Firefox, Safari, IE9+

На рисунку 1.5 представлено веб-сторінку фреймворку UIKit.

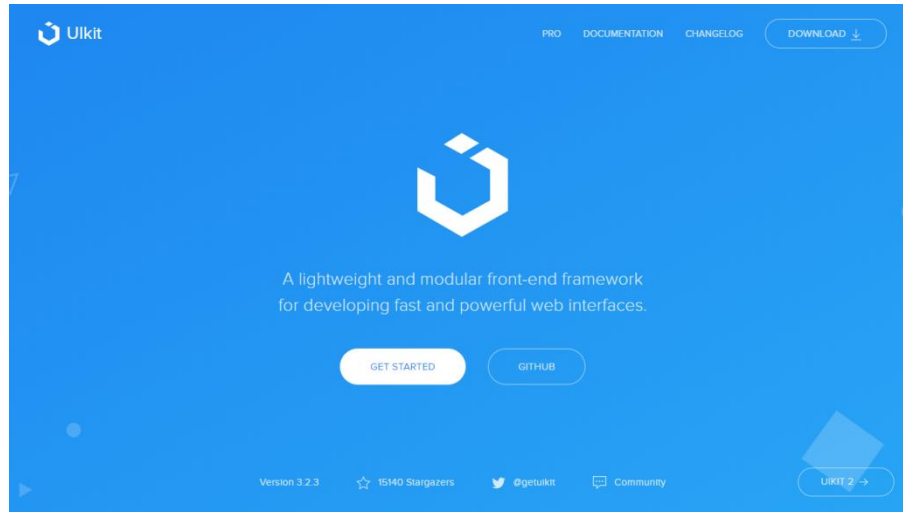


Рисунок 1.5 – Веб-сторінка фреймворку UIKit [43]

Наповнений масою швидких, адаптивних компонентів з узгодженими іменами UIKit став одним з найбільш популярних front-end фреймворків. Він містить більше ніж 30 модульних компонентів які можуть розширюватися і можуть бути об'єднані для більшої універсальності. UIKit містить в собі такі навігаційні компоненти як бокові навігаційні панелі; такі елементи як HTML форми і таблиці; елементи JavaScript такі як off-canvas панелі і модальні діалогові вікна; такі загальні компоненти як кнопки, значки і накладки; і компоненти розмітки включаючи плавну повністю адаптивну систему сітки. [33]

Серед переваг та недоліків фреймворку UIKit можна виділити наступні:

Переваги:

- великі можливості для налаштування під конкретні задачі;
- винятково модульний тож можна додати компоненти до таблиці стилів і не здійснюючи негативного впливу на загальний стиль;
- створення розширених інтерфейсів користувача використовуючи такі компоненти як nestables.

Недоліки:

- дуже мало ресурсів для освоєння фреймворка через його відносну новизну.

UIKit краще підійде для досвідчених розробників через нестачу ресурсів для його освоєння в даний момент. В іншому ж випадку він підходить як для простих так і для складних проектів.[33]

1.4. Задачі дослідження

У відповідності до бажаного кінцевого результату роботи в ній будуть ряд задач. Оскільки головною метою роботи є програмна реалізація інформаційної системи ієрархічної категоризації науковців то більшість цих задач будуть пов'язані з цим. Серед основних задач дослідження можна виділити наступні основні задачі:

- розробка архітектури системи;
- розробка веб-інтерфейсу системи;
- тестування розробленої системи;
- експериментальні дослідження.

Задача розробки архітектури системи полягатиме в визначенні функціоналу системи які вона забезпечуватиме користувачеві. Для візуального представлення функцій системи будуть розроблені UML які дозволять побачити структуру майбутньої системи. Далі відповідно до того виконання яких функцій передбачатиме система буде розроблено структуру системи ієрархічної категоризації науковців і визначити які складові міститиме система. Для візуального представлення структури системи буде створена структурна схема програмного забезпечення системи. Також для представлення принципу функціонування системи будуть розроблені відповідні UML діаграми.

Задача розробки веб-інтерфейсу головним чином полягатиме в розробці інтерфейсу веб-системи ієрархічної категоризації науковців. Для розробки інтерфейсу буде проаналізовано які елементи повинен містити інтерфейс для того щоб забезпечити якісне виконання функцій системи та забезпечення зручного користування системою користувачеві. Після чого з допомогою обраного програмного забезпечення буде реалізовано веб-систему.

Задача тестування розробленої системи полягатиме в перевірці належної роботи самої системи загалом та всього її функціоналу. Зрозумілим є що після розробки будь-якої системи є важливим провести її тестування для того щоб виявити якісь недопрацювання які виникли в процесі розробки або ж виявити помилки які можуть з'являтися при роботі з системою. Також проведення тестування системи може допомогти внести певні зміни які покращитимуть роботу системи і при цьому можливість яких важко було передбачити на етапі розробки.

Задача експериментальних досліджень полягатиме в дослідженні залежностей які існують між показниками цитування науковців. Показники цитування є важливою складовою для системи яка здійснює категоризацію науковців тому дослідження залежностей між показниками цитувань науковців в подальшому може дозволити додати додатковий функціонал до системи що розробляється.

1.5. Висновки до розділу

Інформаційно-пошукові веб-системи зайняли важливе місце у житті людей, а особливо тих людей які є активними користувачами мережі Інтернет. Наразі існує велике різноманіття інформаційно-пошукових систем однак немає таких систем які дозволяли б здійснювати пошук і категоризацію науковців зважаючи на їхню наукову діяльність. Тому в даний момент створення системи ієрархічної категоризації науковців є досить актуальним.

В сучасному світі існує велика маса різноманітних веб-ресурсів тому для забезпечення гідного місця серед них потрібно створити якісний ресурс. Крім того від того наскільки якісно веб-ресурс розроблено залежить те чи користуватиметься даний ресурс популярністю. На даний момент веб-технології сильно розкинулись і відповідно сильно змінилося підходи до створення веб-ресурсів. Уже існує ряд вимог і рекомендацій до того як мають виглядати веб-системи у відповідності до потреб користувачів та у відповідності до існуючих пристроїв на яких той чи інший веб-ресурс

використовуватиметься. Серед найбільш важливих рекомендацій по створенню веб-систем потрібно відмітити такі складові як usability та UX(досвід користування). Від того на скільки ваша веб-система відповідає принципам даних залежить успіх вашої системи у вподобанні її користувачами.

Зважаючи на існуючі сучасні пристрої важливим є щоб веб-система була адаптивною тобто щоб вона без проблем могла відтворюватися на будь-якому пристрої незважаючи на те яке у нього розширення екрану. Для забезпечення адаптивності веб-системи та спрощення її розробки в даний момент доцільно використовувати фреймворк. Оскільки наразі існує багато різних фреймворків то в залежності від того що ви розробляєте і ряду інших вимог потрібно підбирати фреймворк який найкращим чином задовольнятиме ваші потреби.

В даному випадку для розробки системи ієрархічної категоризації науковців було обрано фреймворк Bootstrap 4. Основними причинами його вибору стало те що це надійний фреймворк який дозволяє створювати адаптивні веб-системи і при цьому це найпопулярніший фреймворк в результаті чого існує велика кількість матеріалів та ресурсів для його освоєння що дозволяє почати роботу з ним навіть новачку.

2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-СИСТЕМИ ІЄРАРХІЧНОЇ КАТЕГОРИЗАЦІЇ НАУКОВЦІВ

2.1. Розробка архітектури системи

Розробка будь-якої системи чи програмного забезпечення безумовно потребує попереднього визначення як та чи інша система працюватиме. Тому для внесення ясності у те як функціонуватиме система необхідно розробити архітектуру системи. Архітектура системи дозволяє відобразити те з яких складових частин складається система і як ці частини взаємодіють між собою, і як між цими частинами передається інформація. Відомо що до розробки архітектури системи існують різні підходи кожен з яких дозволяє різними засобами та методами відобразити роботу системи. Для розробки ж архітектури системи у даній роботі буде використаний об'єктний підхід який розуміє під собою моделювання методом UML.

2.1.1. Визначення функцій системи

Перед безпосередньою розробкою системи важливо визначити які функції система буде виконувати. Визначення функцій системи дозволить побудувати функціональну схему системи що в свою чергу дозволить в загальних рисах представити як ця система працює.

В даній роботі розробляється веб-система ієрархічної категоризації науковців що означає від системи вимагатиметься різного роду відображення науковців і впорядкування інформації про них. Зважаючи на це виділимо наступні основні функції системи:

- відображення веб-системи
- пошук науковця;
- сортування науковців;
- фільтр науковців;
- відображення детальної інформації про науковця.

Розглянемо дані функції системи ієрархічної категоризації науковців детальніше.

Функція відображення веб-системи головним чином відповідає за представлення інтерфейсу системи користувачу. В даному інтерфейсі містяться усі компоненти з допомогою яких користувач зможе використовувати інші функції системи. Також ця функція відповідає за відображення деякої початкової інформації яку система отримує з бази даних. Серед цієї інформації головними складовими є звичайно назви галузей та підгалузей які підвантажуються з бази даних для відображення в фільтрах і інформація про всіх науковців яка також підвантажується з бази даних та відображається на сторінці системи.

Функція пошуку науковця відповідає за пошук науковця на основі його імені чи прізвища. Дана функція дозволяє користувачу ввести ім'я науковця у строку пошуку та одразу відобразити йому в підказках імена науковців які підпадають під його пошуковий запит. Далі користувач може обрати серед підказок науковця якого він шукає і отримати інформацію про нього.

Функція сортування містить два варіанта серед яких користувач може обрати. Перша з них розуміє під собою звичайне сортування за одним з трьох показників таких як citations, h-index, i10-index. Обравши один з цих параметрів здійснюється сортування науковців у відповідності до цього показника за всі роки у порядку спадання. Другий варіант сортування являє собою розширене сортування. В даному варіанті є можливість вказати за яким з вище описаних параметрів буде здійснюватися сортування в першу, в другу і в третю чергу. Також разом з порядком сортування за даними параметрами вказується чи сортування буде здійснюватися за показниками за весь час чи за показниками починаючи з 2013 року.

Функція фільтру науковців складається із двох складових пошуку(фільтру) за ключовими словами і фільтру за галузями. Фільтр за ключовими словами дозволяє відобразити серед результатів пошуку науковців які задовольняють введенні користувачем ключові слова. Фільтр за галузями науки дозволяє відобразити науковців які належать до окремої галузі науки або

ж відобразити науковців з декількох галузей. Також окрім галузі науки користувач може задати і відповідні галузі підгалузі науки. І як для галузей так і для підгалузей користувач може задати ступінь належності який вказує який відсоток наукових робіт в науковця в тій чи іншій галузі або підгалузі.

Функція відображення детальної інформації про науковця дозволяє надати користувачеві більш ширший набір інформації про того чи іншого науковця. Ця функція передбачає відображення інформації про обраного науковця в модальному вікні. Серед інформації відображаються індекси цитування не лише за весь час але й за період з 2013 року. Також тут відображаються діаграма належності науковця відповідно до галузей науки і діаграма належності науковця до підгалузей науки.

Функціональну схему системи ієрархічної категоризації науковців у вигляді use-case діаграми показано на рисунку 2.1.

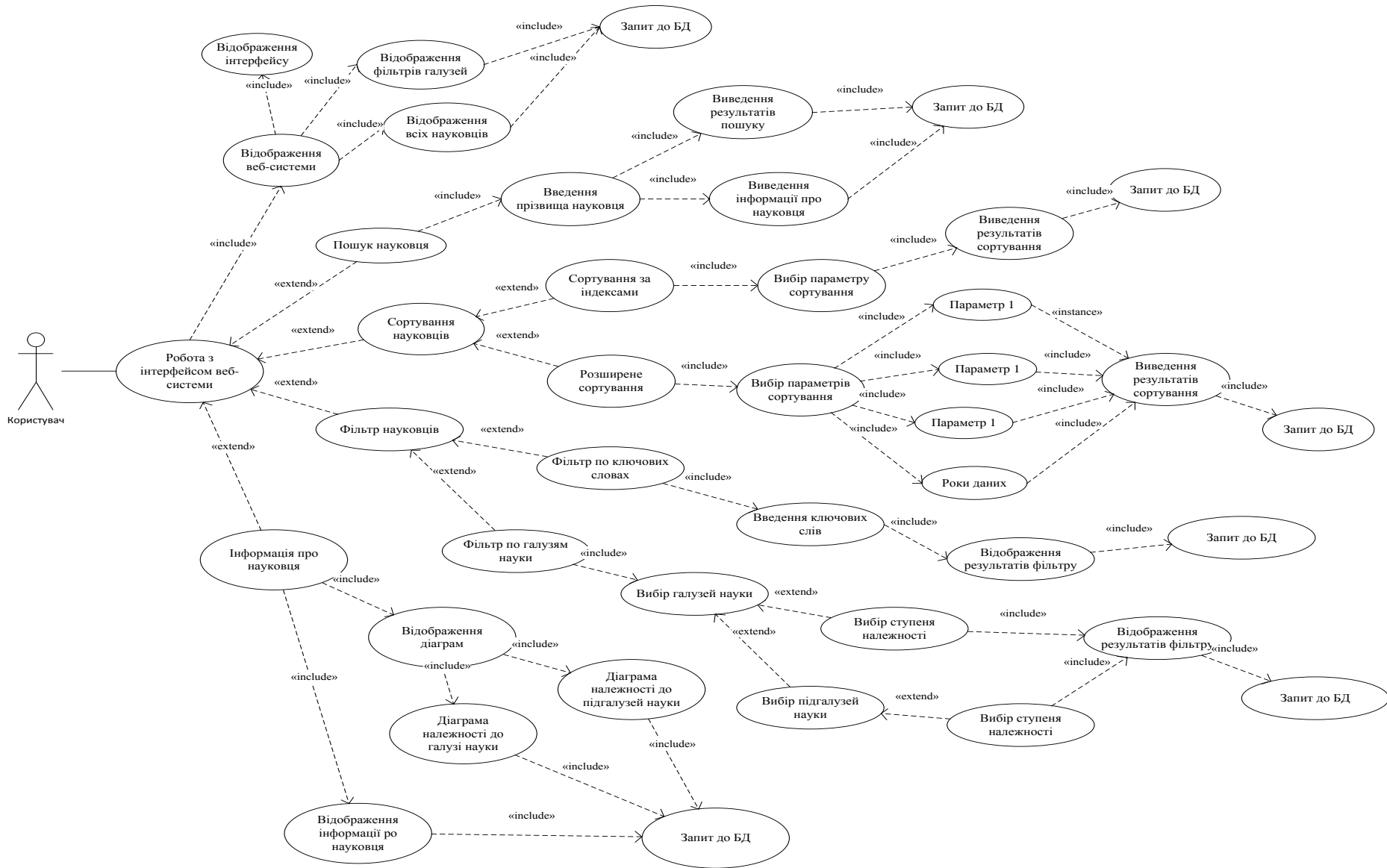


Рисунок 2.1 – Use-case діаграма системи

2.1.2. Розробка алгоритму функціонування системи

При розробці системи також важливо визначити алгоритм за яким працюватимуть ті чи інші її елементи. Система містить вищеописані функції кожна з яких має певний алгоритм роботи.

1) На початку запуску веб-системи виконується ряд дій для того щоб відобразити інтерфейс системи користувачу. Запуск веб-системи містить наступні етапи:

- запуск системи;
- запит до бази даних;
- відображення інтерфейсу;
- відображення списку науковців.

Як тільки користувач відкриває систему в браузері до бази даних надсилається запит на дані для відображення пунктів галузей у фільтрах і на дані необхідні для відображення списку науковців. Далі відбувається відображення інтерфейсу системи і після отримання даних з бази даних відображаються галузі науки та список науковців.

2) Під час використання користувачем системи функції пошуку виконуються етапи:

- зчитування введених даних;
- запит до бази даних;
- виведення результатів пошуку;
- виведення інформації про науковця.

Коли користувач вводить дані до строки пошуку вони автоматично надсилаються до бази даних звідки повертаються дані про науковців які відповідають пошуковому запиту. Ще під час вводу користувачу виводиться підказка з переліком науковців які відповідають пошуковому запиту. Користувач може обрати з цього переліку науковця якщо він відповідає його пошуковому запиту або ж завершити введення пошукового запиту і натиснути пошук. В результаті пошуку користувачу буде виведена інформація про науковця якого він обрав серед результатів пошуку.

3) Якщо користувач хоче відсортувати результати пошуку тоді він може скористатися одним з двох варіантів сортування: сортуванням за індексами цитування або розширеним сортуванням.

Функція сортування за індексами цитування включає наступні етапи:

- вибір параметра сортування;
- запит до бази даних;
- виведення результатів сортування.

В даному випадку після вибору параметру сортування здійснюється запит до бази даних після чого здійснюється обробка даних. Після обробки даних відбувається відображення відсортованих науковців на сторінці системи.

Функція розширеного сортування містить декілька окремих параметрів і відповідно включає наступні етапи роботи.

- вибір першого параметра;
- вибір другого параметра;
- вибір третього параметра;
- вибір відрізка часу даних цитування;
- запит до бази даних;
- виведення результатів сортування.

В данному варіанті сортування серед трьох параметрів сортування обирається за яким з них сортування буде здійснюватися в першу чергу, за яким в другу чергу, і за яким в третю чергу. Крім того обирається чи сортування буде здійснюватися за показниками за весь час чи за показниками в період з 2013 року. Далі здійснюється запит до бази даних після чого дані обробляються і користувачеві виводяться перелік відсортованих науковців.

4) Функція фільтру науковців складається з двох складових пошуку за ключовими словами і фільтру по галузях науки.

Пошук за ключовими словами складається з наступних етапів роботи складається з наступних етапів:

- зчитування введених даних
- запит до бази даних

- виведення відфільтрованих науковців

Коли користувач вводить ключове слово до бази даних надсилається запит після чого дані обробляються і на сторінці системи виводяться відфільтровані науковці.

- фільтр за галузями науки складається з наступних етапів:
- вибір галузі(галузей) науки
- встановлення ступеня належності до галузі
- вибір підгалузі(підгалузей)
- встановлення ступеня належності до підгалузей
- запит до бази даних
- виведення відфільтрованих науковців

Коли користувач обирає галузь науки або декілька галузей науки то до бази даних надсилається запит далі відбувається обробка даних після чого на сторінці системи відображаються відфільтровані науковці. В даній функції системи користувач за бажанням може крім галузі обрати ступінь належності та підгалузі до кожної з обраних галузей, а також може обрати за бажанням до підгалузей також ступінь належності.

5) Ще однією важливою складовою системи є вікно детальної інформації про науковця. Ця функція включає наступні етапи:

- Вибір конкретного науковця
- Запит до бази даних
- Відображення інформації про науковця
- Відображення діаграм належності

В даній функції при виборі конкретного науковця до бази даних надсилається запит на отримання всіх даних про нього. Після отримання даних вони відображаються в модальному вікні. Крім основних даних серед інформації про науковця також відображаються діаграми належності. Одна з діаграм відображає ступінь належності наукових робіт науковця до відповідних галузей науки, а інша відображає ступені належності наукових робіт до відповідних підгалузей науки.

Алгоритм функціонування системи у вигляді діаграми діяльності представлено на рисунку 2.2.

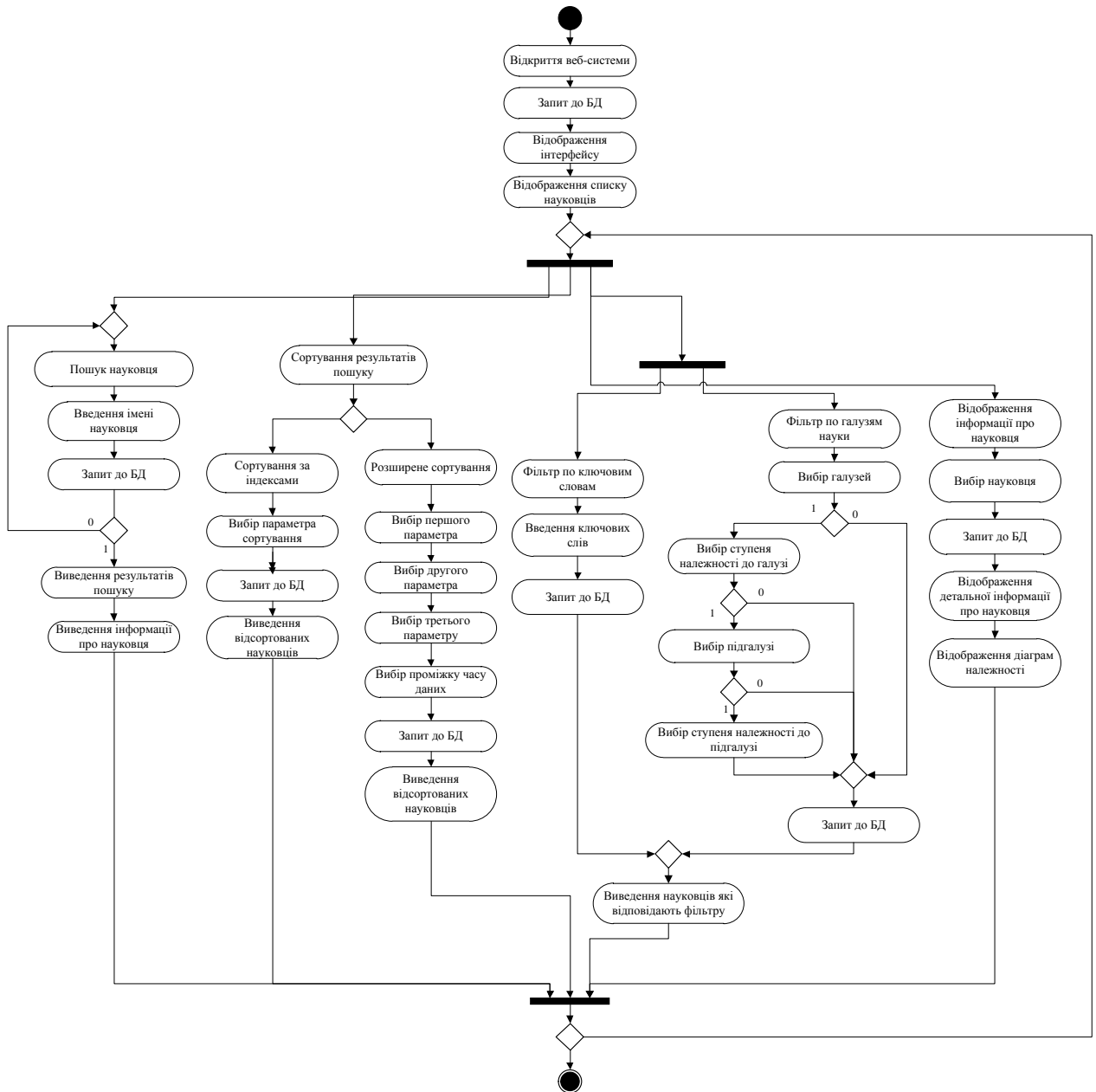


Рисунок 2.2 – Діаграма діяльності системи ієрархічної категоризації науковці

2.2. Розробка структури системи

Перед розробкою системи перш за все потрібно з'ясувати її структуру відповідно до функцій що вона виконує. При розробці будь-якої системи доцільно розділити її на менші підпрограми а бо окремі модулі. Відповідно до цього розробимо структуру системи яка буде складатися з декількох основних модулів.

В даному випадку система буде складатися з наступних модулів:

- модуль інтерфейсу;
- модуль пошуку науковців;
- модуль сортування;
- модуль фільтрів.

Модуль інтерфейсу відповідатиме за відображення системи користувачу. З допомогою цього модулю користувач буде взаємодіяти з системою вводячи певні дані до системи через інтерфейс і після їхньої обробки в інших модулях системи результат з допомогою модулю інтерфейсу відобразатиметься в самому інтерфейсі веб-системи.

Модуль пошуку науковця відповідає за обробку пошукового запиту користувача та передачу результату обробки пошукового запиту на інтерфейс.

Модуль сортування відповідає за забезпечення функції сортування за індексами цитування та функції розширеного сортування. Даним модулем здійснюється обробка параметрів одного з варіантів сортування який обрав користувач та передача результату обробки до інтерфейсу.

Модуль фільтрів відповідає за забезпечення функції фільтрування науковців за ключовими словами та функції фільтрування науковців. Даним модулем здійснюється обробка введених ключових слів користувачем і подальша передача результатів фільтру до інтерфейсу. За схожим принципом здійснюється робота модулю над заданими користувачем галузями.

Структура програмного забезпечення показана на рисунку 2.3.

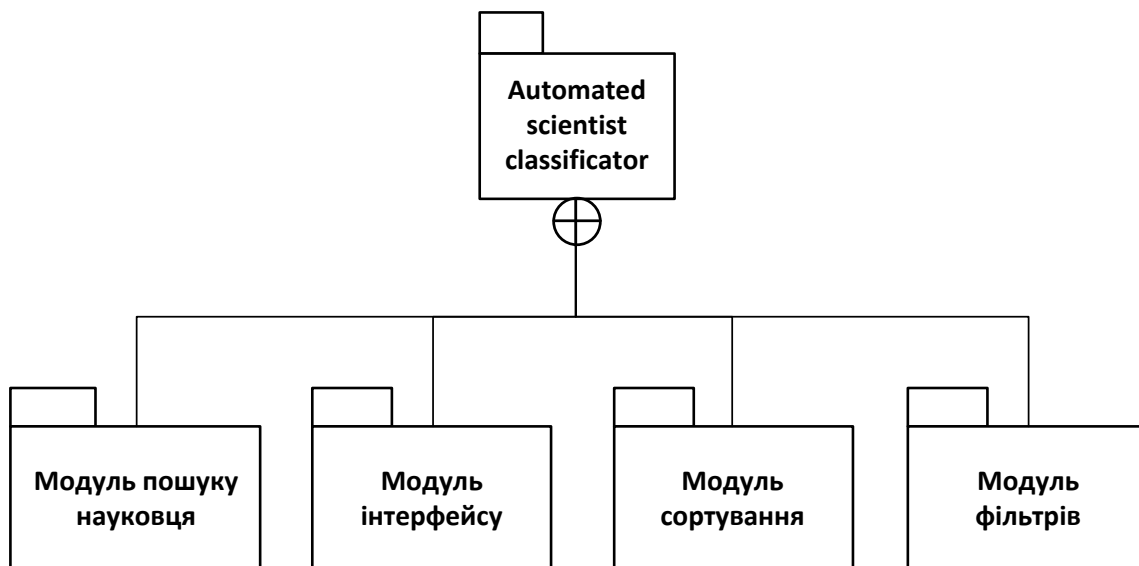


Рисунок 2.3 – Структура програмного забезпечення системи

Щоб представити алгоритм функціонування програмного забезпечення системи скористаємося UML-діаграмою послідовності. Діаграма послідовності – діаграма, на якій зображується взаємодія об'єктів впорядкована в часі. Зокрема, на ній зображуються об'єкти, що беруть участь у взаємодії, і послідовність повідомлень, якими вони обмінюються [44].

UML-діаграма послідовності програмного забезпечення системи представлена на рисунку 2.4

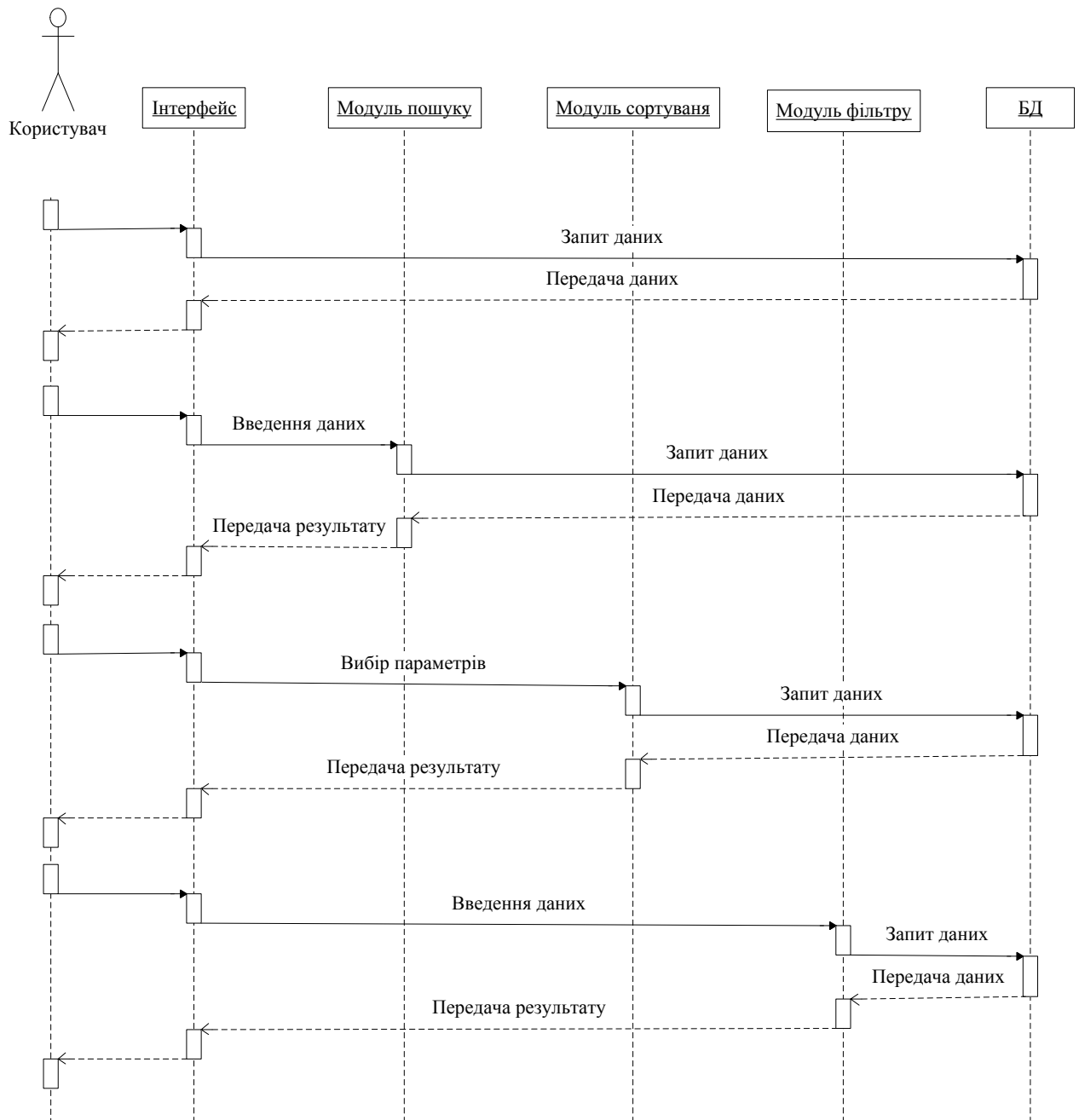


Рисунок 2.4 – UML-діаграма послідовності програмного забезпечення системи представлена на рисунку 2.4

2.2.1. Модуль інтерфейсу системи

Алгоритм роботи модулю інтерфейсу складається з ініціалізації інтерфейсу, запиту до бази даних на отримання даних, відображення елементів інтерфейсу та відображення інформації отриманої з бази даних.

Алгоритм роботи модуля інтерфейсу системи представлено у вигляді блок-схеми на рисунку 2.5.

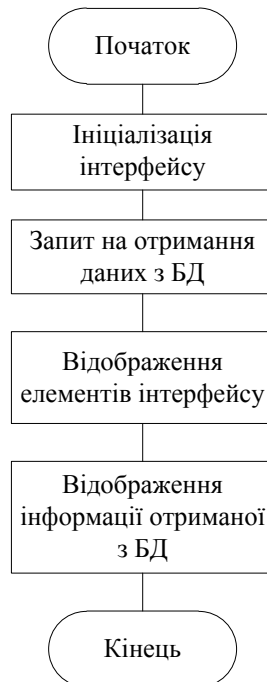


Рисунок 2.5 – Блок-схема алгоритму роботи модуля інтерфейсу системи

Модуль інтерфейсу відповідає за відображення всіх елементів інтерфейсу системи ієрархічної категоризації науковців. При початковому відкритті сторінки веб-системи модуль здійснює запит до бази даних для отримання інформації про всіх науковців список яких початково відображається при відкритті системи. Крім того через цей модуль в інтерфейс відображаються галузі з допомогою яких можна здійснити фільтр перелік яких також отримується з бази даних.

2.2.2. Модуль пошуку

Алгоритм роботи модулю пошуку ієрархічної категоризації науковців складається з отримання пошукового запиту з модуля інтерфейсу, запиту до бази даних, обробки даних і передачі даних на модуль інтерфейсу.

Алгоритм роботи модуля пошуку представлено у вигляді блок-схеми на рисунку 2.6



Рисунок 2.6 – Блок-схема алгоритму роботи модуля пошуку системи

Модуль пошуку ієрархічної категоризації науковців відповідає за надання користувачу системи можливості здійснити пошук конкретного науковця. Після отримання пошукового запиту від модуля інтерфейсу в модулі відбувається запит до бази даних далі відбувається обробка даних під час якої відбираються релевантні результати пошуку. Далі відбувається формування результатів пошуку у формат з якого в подальшому через модуль інтерфейсу користувачу буде представлено список підказок у відповідності до його пошукового запиту.

2.2.3. Модуль сортування

Алгоритм сортування модуля сортування системи ієрархічної категоризації складається з отримання параметрів сортування від модуля інтерфейсу, формування запити до БД, запит до БД, обробку даних і передачу даних на модуль інтерфейсу.

Алгоритм роботи модуля сортування системи ієрархічної категоризації науковців представлено у вигляді блок-схеми на рисунку 2.7.



Рисунок 2.7 – Блок-схема алгоритму роботи модуля сортування системи

Модуль сортування ієрархічної категоризації науковців відповідає за здійснення сортування результату пошуку науковців за одним з наявних варіантів. Після вибору користувачем параметрів сортування з модуля інтерфейсу до модуля сортування передаються які саме параметри користувач обрав. В залежності від того який варіант сортування користувач обрав модуль сортування формує запит і надсилає його до бази даних. Отримані дані обробляються після чого вони передаються на модуль інтерфейсу через який вони в подальшому будуть відображені в відсортованому переліку науковців.

2.2.4. Модуль фільтрів

Алгоритм роботи модуля фільтр системи ієрархічної категоризації науковців складається з отримання показника фільтра від модуля інтерфейсу, формування запиту і запиту до БД, обробку даних і передачу даних на модуль інтерфейсу

Алгоритм роботи модуля фільтр системи представлено у вигляді блок-схеми на рисунку 2.8.



Рисунок 2.8 – Блок-схема алгоритму роботи модуля

Модуль фільтрів відповідає за надання користувачеві функції фільтру науковців за ключовими словами і функції фільтру науковців за галузями та підгалузями науки. Користувач може задати фільтри за бажанням тому з модуля інтерфейсу на модуль фільтрів можуть надійти різні параметри. В залежності від того які параметри надійшли модуль формує і надсилає запит до бази даних. Далі здійснюється обробка даних і подальша їхня передача на модуль інтерфейсу через який вони вже відображаються в інтерфейсі.

2.3. Проектування інтерфейсу

2.3.1. Проектування інтерфейсу веб-сторінок системи

Для забезпечення взаємодії користувача з інформаційною системою ієрархічної категоризації науковців потрібно розробити веб-інтерфейс. Важливою властивістю інтерфейсу має бути його відповідність функціям і вимогам до системи. Крім того інтерфейс веб-системи має бути зрозумілим для користування і містити елементи які забезпечуватимуть легкий доступ користувача до функцій системи.

Для забезпечення виконання функцій системи було вирішено розділити інтерфейс на такі складові як головну сторінку веб-системи і модальне вікно у якому буде відображатися детальна інформація про обраного науковця.

Веб-інтерфейс системи представляє собою веб-сторінку функції якого дозволяють здійснити категоризацію науковців. Інтерфейс веб-сторінки дозволяє за відкритими даними Google Scholar здійснити пошук окремого науковця чи науковців, також отримати необхідну інформацію про них. Загальний вигляд головної сторінки веб-інтерфейсу системи ієрархічної категоризації науковців представлено на рисунку 2.9.

Рисунок 2.9 – Головна сторінка веб-інтерфейсу інформаційної системи

Зважаючи на те що система являє собою веб-сторінку то відповідно вона містить усі основні складові елементи які має містити сучасна веб-сторінка такі як навігаційне меню вгорі та внизу сторінки, пошук, функції сортування та фільтру, а також пагінацію.

У відповідності до рекомендацій з usability до веб-інтерфейсів елементи сторінки були розміщені за загальноприйнятими стандартами таким чином як показано на рисунку 2.9.

А якщо бути більш конкретним то навігаційне меню веб-сторінки як і прийнято було розташовано вгорі з наявними у ньому імені та логотипу ресурсу і пунктами меню.

Строку пошуку різні ресурси розміщують по різному однак зазвичай вона знаходиться вгорі веб-сторінки. В даному випадку вона також була розташована вгорі одразу після навігаційного меню щоб користувач при користуванні системою міг як найшвидше отримати доступ до функції пошуку.

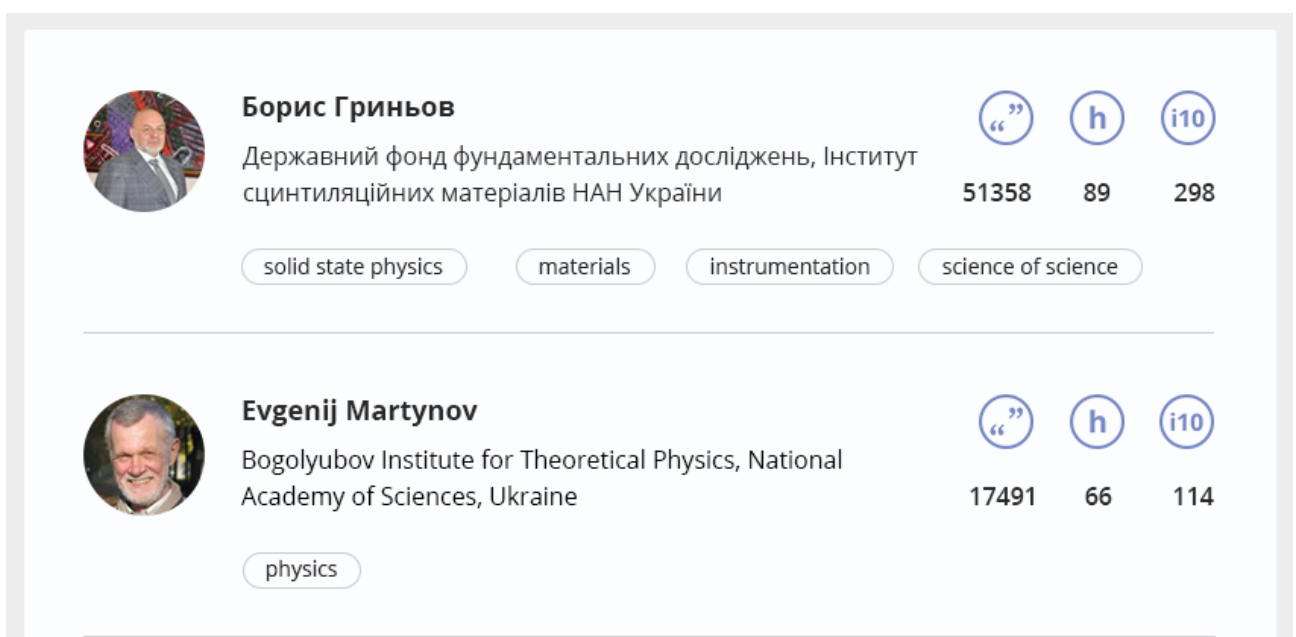
Блок фільтрів також має більшу варіативність в розміщенні в порівнянні з навігаційним меню. Він може розташовуватися в лівій чи правій частині веб-

сторінки також його деколи розташовують вгорі. В даному випадку було вирішено розмістити блок фільтрів в лівій частині веб-сторінку зважаючи на те що таке розміщення обирають більшість ресурсів і відповідно таке розміщення буде зрозумілим і зручним користувачам.

Елементи сортування було розміщено над головним контентом(результатами пошуку) веб-сторінки зважаючи на те що таке розміщення також є загально прийнятим і тому воно буде зрозумілим користувачам системи. Безумовно можуть існувати інші варіанти розміщення елементів сортування однак рішення про їх розміщення було прийнято зважаючи на usability та на користувачів системи які знайомі саме з таким їх розміщенням.

Пагінація також зважаючи за загальноприйняті стандарти було розміщено внизу веб-сторінки одразу після основного контенту.

За замовчуванням система відображає в результатах пошуку всіх науковців які містяться в базі даних. Результати пошуку представлені у вигляді блоків які містять таку інформацію про науковців як ім'я, заклад де він працює, його наукові інтереси, та індекси цитування. Вигляд представлення результатів пошуку науковців представлено на рисунку 2.10.



Рисунк 2.10 – Вигляд представлення результатів пошуку науковців

Серед індексів цитування представлені такі індекси як сумарний обсяг цитування, індекс Хірша(h-індекс) та i10-індекс. У відповідній послідовності вони представлені в результатах пошуку. З метою покращення usability інтерфейсу позначення значення конкретного індексу було представлено з допомогою іконок які лаконічно інформують користувача якому саме індексу відповідає конкретне значення. Вигляд представлення індексів цитування представлено на рисунку 2.11.

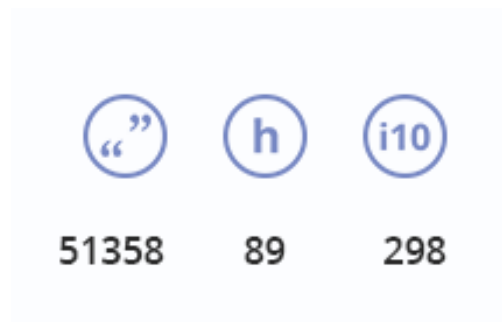
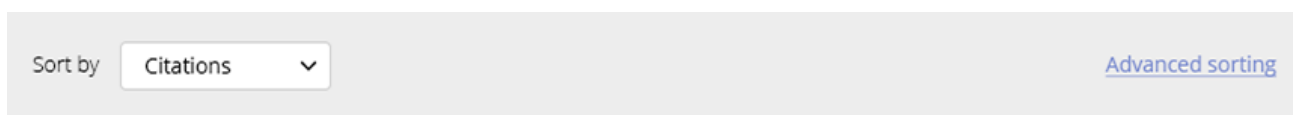


Рисунок 2.11 – Вигляд представлення індексів цитування науковця

При пошуку науковця індекс цитування є важливим аспектом тому було до інтерфейсу системи було додано можливість сортування за одним з трьох індексів цитування. За замовчуванням у користувача є можливість швидко відсортувати результати пошуку саме за одним з трьох індексів. Елементи веб-інтерфейсу з допомогою яких здійснюється сортування за індексами цитування детальніше показано на рисунку 2.11.



Рисунку 2.11 – Елементи веб-інтерфейсу функції сортування

Поруч із самим елементом сортування який розташований зліва на вище представленому рисунку також справа розташовується кнопка яка дозволяє показати елементи розширеного сортування.

Для забезпечення функції розширеного сортування до веб-інтерфейсу було додано чотири випадючих списків з допомогою яких користувач може задати параметри сортування, і таким чином задати за яким індексом буде відбуватися сортування в першу, в другу і в третю чергу, і крім того користувач може обрати сортування або з усіх результатів пошуку або сортування відповідно до цитування науковця лише за останні 5 років. Також додана кнопка після натиснення якої відбувається сортування. Вигляд елементів інтерфейсу які забезпечують функцію сортування результатів пошуку представлено на рисунку 2.12.

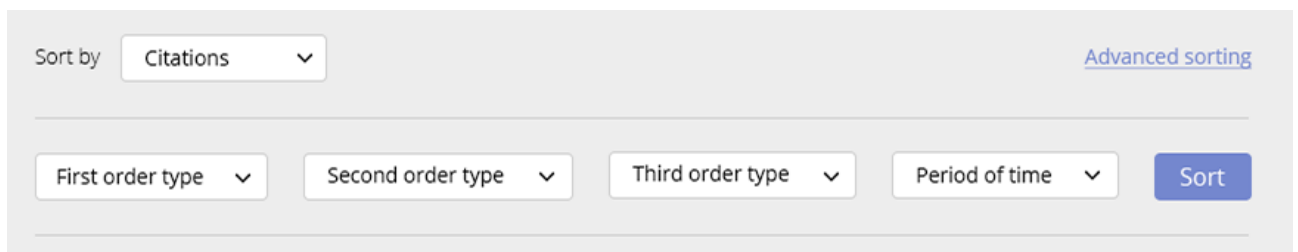


Рисунок 2.12. – Вигляд елементів інтерфейсу функції сортування

Для того щоб можна було відобразити в результатах пошуку науковців за галуззю їхньої наукової діяльності існує функція фільтру. Загалом інтерфейс системи містить такі фільтри як пошук за ключовими словами та фільтр за галуззю наукової діяльності. Функція пошуку за ключовими словами здійснює фільтр науковців відповідно до конкретних наукових інтересів. Фільтр за галуззю дозволяє відобразити в результатах пошуку науковців які працюють в конкретній галузі чи підгалузі науки. Також є функція яка дозволяє відобразити науковців з певним рівнем належності до певної галузі. Зважаючи на це в блок фільтрів було додано елементи які забезпечать зручне виконання вище описаних функцій.

Відповідно в блоці фільтрів є назви фільтрів як дозволять користувачеві легко орієнтуватися в цій функції системи. Також поруч з назвами фільтрів було розміщено кнопки «Clear» з допомогою яких користувач зможе швидко очистити той чи інший фільтр.

Для забезпечення функції фільтру за ключовими словами було додано строку вводу в яку користувач зможе ввести ключове слово. Після введення ключового слова користувач зможе бачити які ключові слова він додав тому вони будуть представлені нижче строки вводу у вигляді блоків. Натиснувши також на відповідній кнопці цих блоків у користувача також буде можливість видалити з фільтру окреме ключове слово.

Для забезпечення функції фільтру науковців за галузями було додано ряд прапорців(checkbox) із назвами галузей які користувач може відмітити і відповідно отримати в результаті перелік науковців які належать певної галузі чи галузей. Також після відмітки прапорця галузі в інтерфейсі з'являтимуться прапорці підгалузей, а також повзунок з допомогою якого можна встановити ступінь належності науковців до обраної галузі. Такі ж повзунки з'являтимуться і при відмітці прапорців підгалузей.

Вигляд представлення функції фільтру результатів пошуку в інтерфейсі системи представлено на рисунку 2.13

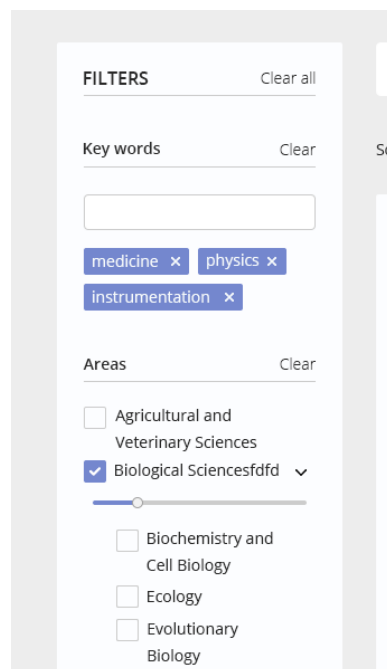


Рисунок 2.13 – Вигляд представлення функції фільтру результатів пошуку

При виборі в результатах пошуку конкретного науковця відкривається вікно з більш детальною інформацією про науковця. В цьому вікні зокрема

відображається посилання на профіль науковця в Google Scholar, діаграма належності науковця до галузей науки і діаграма належності науковця до підгалузей науки.

На діаграмах відображається відсоткове співвідношення наукової діяльності науковця у певній галузі та підгалузі науки. Індекси цитування в цьому вікно представлені в розширеному вигляді де крім загальних значень також ще представлені значення цитувань за останні 5 років. Інтереси науковця, показники цитування і діаграми в даному вікні представленні окремими блоками кожен з яких має відповідний заголовок. Дане представлення дозволить користувачу легко знайти потрібну йому інформацію. Вигляд вікна детальної інформації про науковця представлено на рисунку 2.14.

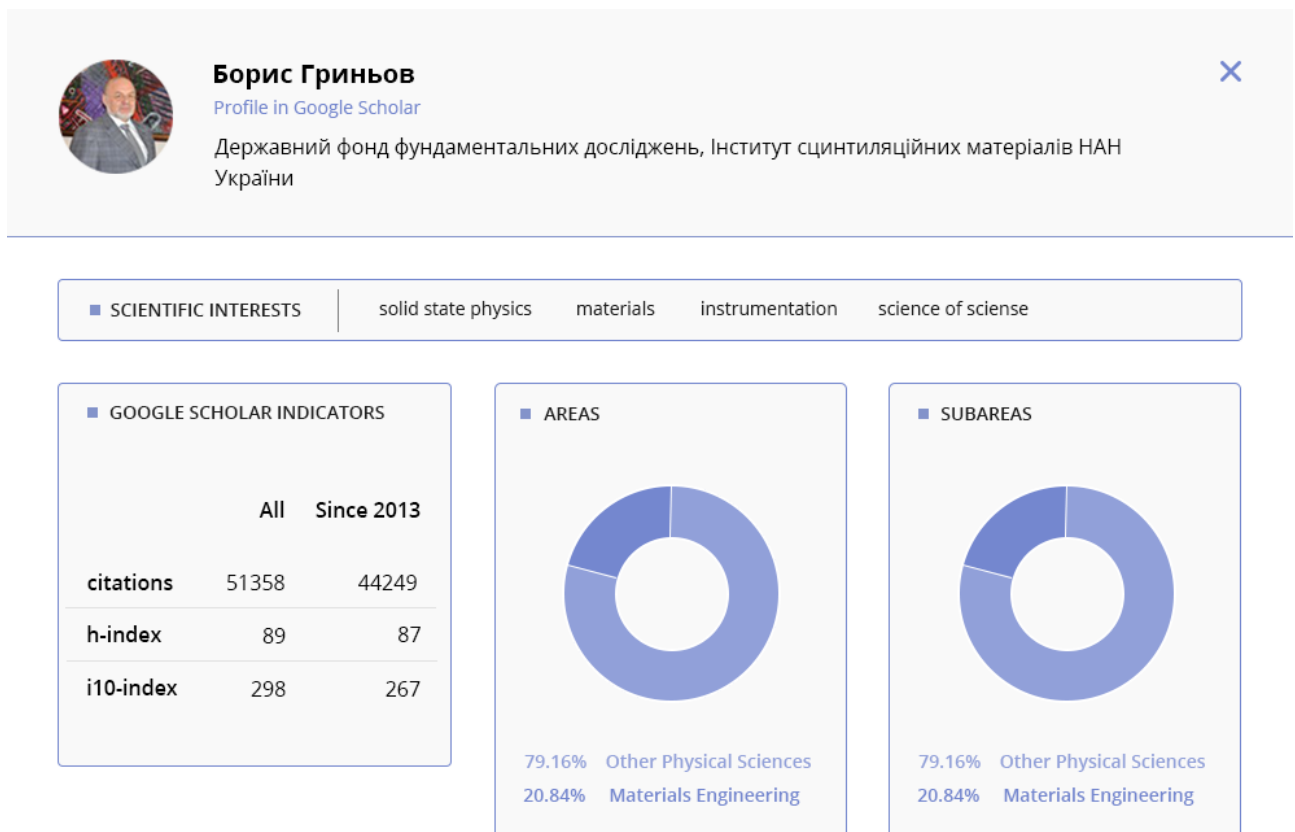


Рисунок 2.14 – Вигляд вікна детальної інформації про науковця

2.3.2. Дослідження математичних підходів до визначення показників usability веб-системи

Юзабіліті веб-ресурсу можна досягнути з допомогою дотримання різноманітних рекомендацій до щодо розміщення елементів інтерфейсу, правильного підбору кольорів та шрифтів та дотримуючись послідовності в дизайні сайту. Однак дані складові є доволі абстрактними і їх важко чітко оцінити. Однак існують показники юзабіліті так звані «usability metrics» кожен з яких можна математично обрахувати зважаючи на результати тестування. ISO/IEC 9126-4 Metrics рекомендує щоб показники юзабіліті включали наступне:

- ефективність: точність та повнота з якою користувачі досягають конкретних цілей;
- продуктивність: ресурси витрачені у співвідношенні точності і повноти, з якою користувачі досягають цілей;
- задоволення: комфорт та прийнятність використання [45].

Щоправда фактичні способи їх обрахунку часто залишаються на розгляд оцінювача. Розгляньмо розрахунок кожного з показників.

Ефективність може бути обрахована з допомогою виміру відсотку завершення. У відповідності до фундаментальних показників юзабіліті відсоток завершення може бути обрахований призначенням бінарного значення '1' якщо учаснику тесту вдається виконати завдання і значенням '0' якщо не вдається [46].

Таким чином ефективність може бути представлена у вигляді відсотку використовуючи наступну формулу [VV]:

$$E = \frac{N_s}{N_u} * 100\% \quad (2.1)$$

де E – показник ефективності,

N_s – число завдань які було завершено успішно,

N_u – загальне число виконаних завдань.

Хоча даний показник повинен наближатися до 100% однак у відповідності до дослідження проведеним Джефом Сауро середній відсоток завершення становить 78%. [46]

Розглянемо до прикладу випадок коли 5 користувачів виконують задачу використовуючи одну й ту саму систему. По завершенні тестування 3 користувачі досягають цілі завдання і той час як 2 інших ні. Зважаючи на дані показники проведемо обрахунок.

$$E = \frac{3}{5} * 100\% = 60\%$$

Показники юзабіліті для продуктивності вимірюються з точки зору часу який потребує учаснику тестування на виконання завдання. Продуктивність може бути обрахована у 3 способи: продуктивність на основі часу, загальна відносна продуктивність та експертна продуктивність. Продуктивність яка базується на часі може бути обрахована за наступною формулою [47]:

$$\bar{P}_t = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{NR} \quad (2.2)$$

де \bar{P}_t – показник продуктивності за часом,

N – це число сценаріїв (завдань),

R – це кількість учасників тестування,

n_{ij} – це результат проходження завдання і учасником j; $n_{ij}=1$ якщо сценарій був завершений успішно і учасник досягнув цілі, і $n_{ij}=0$ якщо сценарій був неуспішним і учасник провалив досягнення цілі,

t_{ij} – це час витрачений учасником j щоб пройти через сценарій i. У випадку неуспішного виконання сценарію час вимірюється до моменту виходу з сценарію учасником в результаті відмови від цілі чи виходу з системи.

Варто зауважити що на відмінну від ефективності, продуктивність на базі часу є абсолютною величиною а не відносною.

Загальна відносна продуктивність використовує співвідношення часу який був затрачений користувачами які успішно завершили задачу до загального часу затраченого всіма користувача. Формула обрахунку загальної відносної продуктивності може бути представлена наступним чином [47]:

$$\bar{P} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} * 100\% \quad (2.3)$$

Можна також розрахувати додаткову відносну ефективність продукту, так звану експертну продуктивність. В даному випадку окрім користувачів які належать до цільової аудиторії в тестуванні також бере участь експерт. Експерт може бути розробник системи або ж користувач який добре обізнаний в архітектурі і тому може пройти сценарій з максимальною можливою швидкістю користувача і апріорним успішним завершенням. Експертна продуктивність на базі часу обраховується за наступною формулою [47]:

$$\bar{P}_{te} = \frac{\sum_{i=1}^N \frac{1}{t_{0i}}}{N} \quad (2.4)$$

де t_{0i} – це ідеальний час який потрібен експерту для завершення сценарію і

Фізичний сенс експертної продуктивності за часом полягає в максимальні теоретично можливій швидкості роботи з продуктом.

Відповідно відносну продуктивність за часом можна обрахувати за формулою [47]:

$$\bar{P}_E = E * \frac{R \sum_{i=1}^N t_{0i}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \quad (2.5)$$

де t_{0i} – це ідеальний час який необхідний експерту щоб завершити сценарій i ,

E – це загальна ефективність системи.

Фізичний сенс відносної експертної продуктивності є мірою потенційної продуктивності по відношенню до фактичної системи з точки зору її користувацької ефективності.

Статистична помилка обраховуватиметься за формулою [47]:

$$\sigma_P = \sqrt{\frac{\bar{P}(10 - \bar{P})}{R}} \quad (2.6)$$

Загальну продуктивність продукту можна обрахувати за формулою [47]:

$$P = \bar{P} \pm \sigma_P \quad (2.7)$$

Розглянемо випадок що 4 користувачі працюють з продуктом у відповідності до одного сценарію. Три користувачі завершують сценарій успішно і один не успішно. Час завершення сценарію першим користувачем складає 1 секунду, другим користувачем 2 секунди, третім користувачем 3 секунди і 10 секунд четвертим користувачем який завершив сценарій не успішно. Експерт завершує сценарій за 1 секунду. Відповідно до цього обрахуємо показник продуктивності в часі, показник загальної відносної продуктивності та показник експертної продуктивності для даного випадку.

$$P_t = \frac{\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{0}{6}\right)}{1 * 4} = \frac{11}{24} \text{ (задач / сек)}$$

$$P = \frac{(1 * 1 + 1 * 2 + 1 * 3 + 0 * 10)}{1 + 2 + 3 + 10} * 100\% = 37.5\%$$

$$P_e = 75\% * \frac{4 * 1}{1 + 2 + 3 + 10} = 18.75\%$$

На відміну від ефективності і продуктивності де ми спостерігаємо і оцінюємо дії користувача, показник задоволення направлено на суб'єктивні думки користувача. Показник задоволення вимірюється за допомогою

формалізованої анкети із вказанням шкал задоволеності. Кожна анкета містить число стверджень які відображають суб'єктивну думку користувача на взаємодію з системою. Середній показник задоволення продуктом оцінюється за формулою [48]:

$$\bar{S} = \left(\frac{\sum_{j=1}^R \sum_{i=1}^{Q_+} \frac{P_{ij}^+}{4} + \sum_{j=1}^R \sum_{i=1}^{Q_-} \frac{P_{ij}^-}{4}}{(Q_+ + Q_-)R} \right) * 100\% \quad (2.7)$$

де R – число респондентів,

Q_+ – число позитивних запитань представлених в анкеті,

Q_- – число негативних питань,

P_{ij}^+ – вага відповіді на позитивне запитання (0-4) для i -сценарію та j -респондент,

P_{ij}^- – вага відповіді на негативне запитання (0-4) для i -сценарію та j -респондент.

2.4. Висновки до розділу

В даному розділ було проведено роботу над розробкою архітектури системи яка полягала в визначенні функцій та структури майбутньої системи. Даний етап розробки був важливий для наявності в подальшому чіткого розуміння того що собою являє система, яким чином вона працює та з яких компонентів. Розроблена архітектура в подальшому допомогла легко розпочати роботу над інтерфейсом системи.

Після визначення архітектури системи було розпочато роботу над інтерфейсом веб-системи. Зважаючи на функції які має виконувати система було вирішено які елементи має містити інтерфейс веб-системи. Зважаючи на загальноприйняті принципи розміщення елементів інтерфейсу веб-системи було розроблено інтерфейс який відповідає сучасним вимогам до веб-систем.

Тому розроблений інтерфейс дозволяє забезпечити зручне та легке користування функціями веб-системи користувачем.

3 ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Тестування є важливим етапом розробки будь-якої системи тому що на даному етапі відбувається перевірка роботи системи з метою переконання що всі елементи системи працюють правильно ніде не з'являється ніяких помилок. Крім того тестування також може проводитися з метою помилок в системі і подальшого їхнього усунення. Також на етапі тестування можна попрацювати з уже готовим інтерфейсом системи і що може допомогти знайти різні рішення для покращення інтерфейсу. Тестування інтерфейсу веб-системи ієрархічної категоризації науковців буде проводитися по наступних критеріях:

- пошук науковця за прізвищем;
- пошук науковця за спеціальностями;
- пошук науковця за галузями;
- рейтингування науковців.

За кожним з критеріїв буде проводитися тестування певного функціоналу системи за який відповідає той чи інший критерій. Розгляньмо кожен критерій тестування детальніше

3.1. Пошук науковця за прізвищем

Функція пошуку науковця здійснюється за допомогою поля вводу яке знаходиться вгорі сторінки веб-системи та кнопки «Search» яка знаходиться праворуч від нього. Поле вводу функції пошуку містить підказку яка вказує на те яку саме інформацію потрібно ввести в нього. Поле пошуку до початку введення пошукового запиту показано на рисунку 3.1.

The image shows a horizontal search bar. On the left, there is a text input field with a light gray border and a placeholder text 'Scientist name'. To the right of the input field is a blue rectangular button with the word 'Search' written in white text.

Рисунок 3.1 – Вигляд поля пошуку до введення пошукового запиту

Одразу після того як користувач почав вводити ім'я майже одразу йому виводиться підказка з списком науковців які відповідають щойно введеному ним пошуковому запиту. Після чого користувач може продовжити введення пошукового запиту або ж одразу вибрати потрібного науковця зі списку. Вигляд елементів пошуку під час введення пошукового запиту показано на рисунку 3.2.

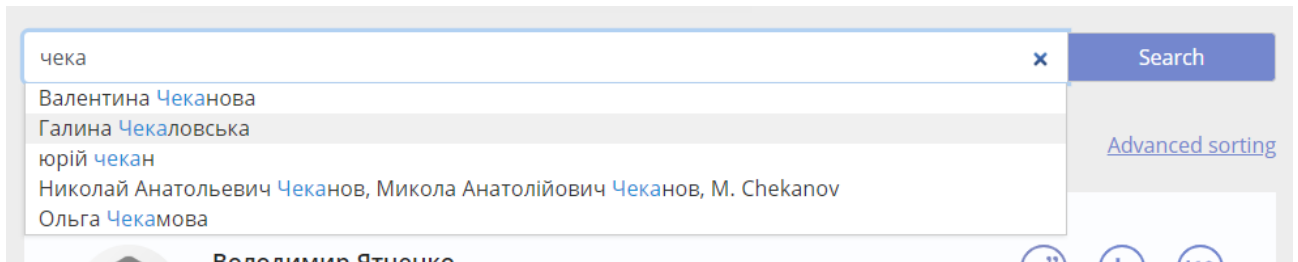


Рисунок 3.2 – Вигляд елементів пошуку під час введення пошукового запиту

Після того як користувач обрав одного з науковців імена яких йому пропонувалися у підказці ім'я цього науковця поміщується в строку пошуку так як показано на рисунку 3.3.

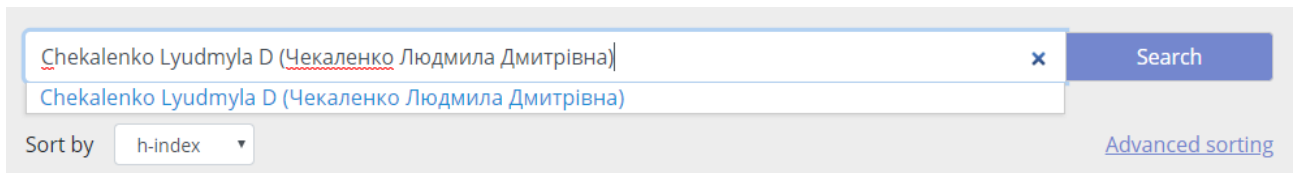


Рисунок 3.3 – Вигляд елементів пошуку після вибору науковця зі списку у підказці

Коли ім'я конкретного науковця поміщене у строку пошуку після натиснення кнопки «Search» користувачу виводиться модальне вікно з детальною інформацією про науковця у такому вигляді як показано на рисунку 3.4.

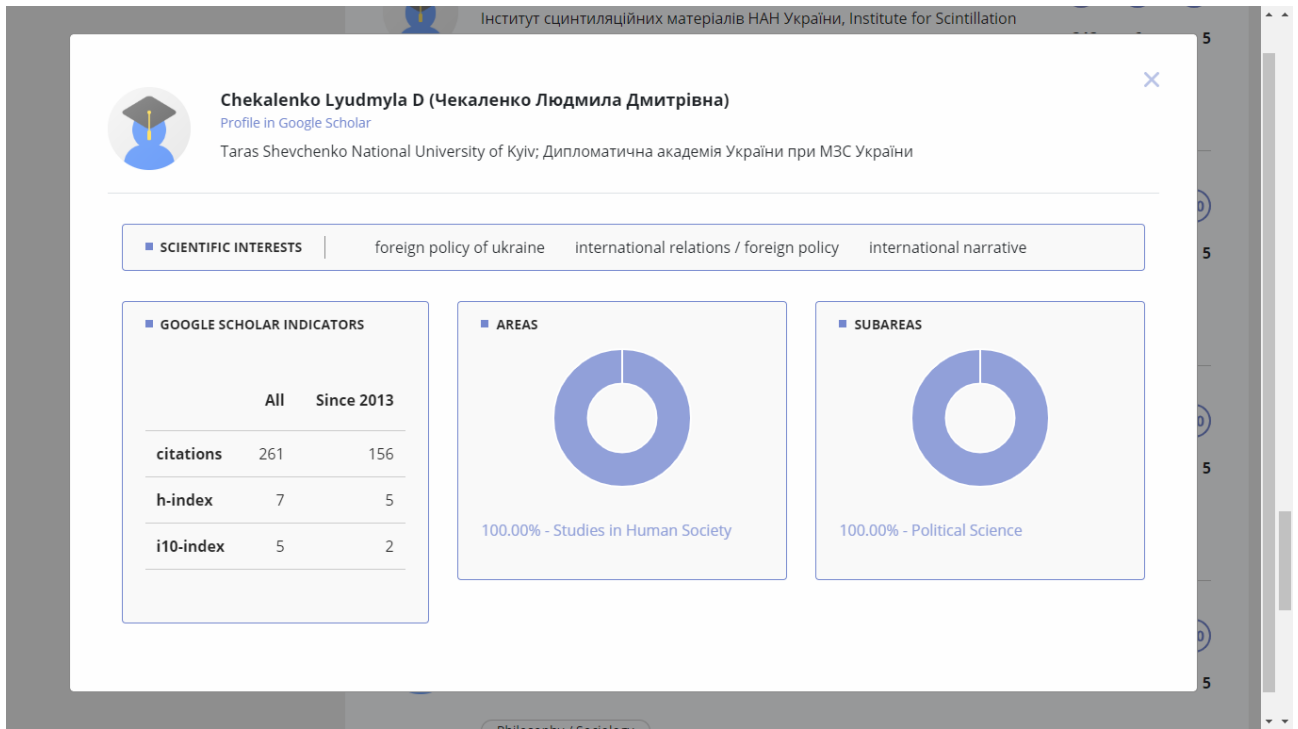


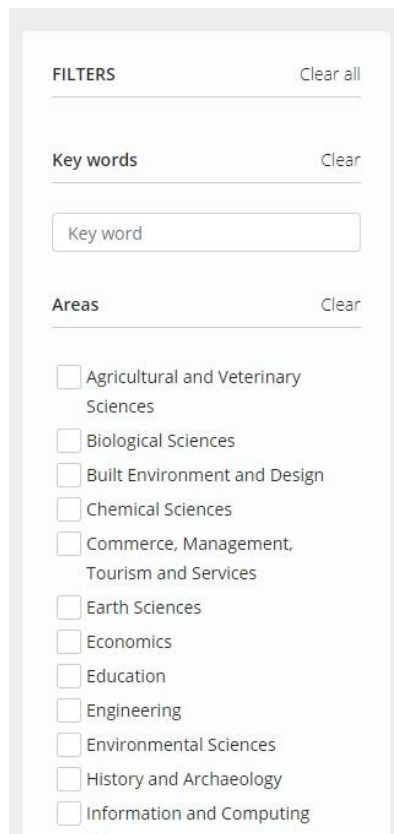
Рисунок 3.4 – Модальне вікно з детальною інформацією про обраного науковця

В даному модальному як і описувалося раніше міститься різна загальна інформація окрім якої можна виділити посилання на сторінку науковця на Google Scholar, блок у якому представлено наукові інтереси науковця, блок з детальними показниками цитування, індексів Хірша та i10-індексів, а також дві діаграми одна з яких показує ступінь належності науковця до галузей науки в той час як інша показує ступінь належності науковця до підгалузей науки.

Крім відображення в результаті пошуку дане модальне вікно відображається і тоді коли користувач натискає на певного науковця і серед результатів пошуку чи сортування.

3.2. Пошук науковців за галузями

За замовчуванням після відкриття веб-системи у ній відображаються в блоці результатів пошуку відображаються всі науковці. Тому для того щоб здійснити пошук за галузями науки користувач має відмітити у блоці фільтрів прапорців поруч з галуззю чи декількома галузями за якими він хоче знайти науковців. Початково блок фільтрові виглядає так як показано на рисунку 3.5.



The image shows a search filter interface with the following elements:

- FILTERS** (header) with a **Clear all** link.
- Key words** section with a **Clear** link and an empty text input field labeled "Key word".
- Areas** section with a **Clear** link and a list of 13 categories, each with an unchecked checkbox:
 - Agricultural and Veterinary Sciences
 - Biological Sciences
 - Built Environment and Design
 - Chemical Sciences
 - Commerce, Management, Tourism and Services
 - Earth Sciences
 - Economics
 - Education
 - Engineering
 - Environmental Sciences
 - History and Archaeology
 - Information and Computing

Рисунок 3.5 – Вигляд елементів фільтру у початковому стані

Після того як користувач відмічає прапорець біля однієї з галузей науки відбувається сортування і в результаті в результатах пошук з'являються науковці які належать до обраної галузі. Крім того під назвою галузі з'являється повзунок який дозволяє встановити ступінь належності у відповідності до якого будуть чи не будуть відображатися в результатах пошуку певні науковці. Вигляд сторінки веб-системи після відмітки галузі науки показано на рисунку 3.6.

The screenshot shows the ASC (Automated Scientist Classifier) web interface. At the top, there is a navigation bar with 'ASC AUTOMATED SCIENTIST CLASSIFICATOR' on the left and 'ABOUT US', 'HOW IT WORKS', and 'RESPONSE' on the right. Below the navigation bar, there is a search bar labeled 'Scientist name' with a 'Search' button. To the left of the search bar is a 'FILTERS' section with 'Clear all' and 'Key words' with 'Clear'. Below the search bar, there is a 'Sort by' dropdown menu set to 'h-index' and a link for 'Advanced sorting'. The main content area displays a list of scientists, each with a profile picture, name, affiliation, and citation metrics (citations, h-index, and i10-index). The first scientist is Vitaly Grishchenko, V.H. Грищенко, with 1874 citations, an h-index of 16, and an i10-index of 51. The second scientist is Василь Чумак, with 46 citations, an h-index of 4, and an i10-index of 2. The third scientist is Andriy Babytskiy, with 7 citations, an h-index of 2, and an i10-index of 0. A filter for 'Agricultural and Veterinary Sciences' is selected, and a 'Low' indicator is visible next to it.

Scientist Name	Affiliation	Citations	h-index	i10-index
Vitaly Grishchenko, В.Н. Грищенко	Kaniv Nature Reserve, Ukraine; Каневский природный заповедник	1874	16	51
Василь Чумак	Ужгородський національний університет, біологічний факультет	46	4	2
Andriy Babytskiy	НУБІП України	7	2	0

Рисунок 3.6 – Вигляд сторінки веб-системи після застосування фільтру по галузях

3.3. Пошук науковців за спеціальностями

Після того як користувач відмітив прапорець певної галузі під назвою даної галузі з'являється не лише повзунок належності до галузі науки а також там з'являється список підгалузе кожна з яких можна обрати для пошуку натиснувши на прапорець поруч.

Так само як і з галузями після відмітки прапорця підгалузі нижче її назви з'являється повзунок з допомогою якого можна встановити на показник у відповідності до якого в результатах пошуку будуть відображатися науковці.

Вигляд сторінки веб-системи після застосування фільтру за підгалузями науки показано на рисунку 3.7.

The screenshot shows the ASC web interface. On the left, there are filters for 'Key words' and 'Areas'. Under 'Areas', 'Agricultural and Veterinary Sciences' is selected, and 'Animal Production' is further selected. The search results list three scientists:

Scientist Name	Citations	h-index	i10-index
Юлія Пославська Julija Poslavska Stepan Gzhytskyi National University of Veterinary Medicine and Biotechnologies Lviv	28	3	0
Постернак Леонід Іванович / Posternak Leonid / Постернак Л. І. Вінницький національний аграрний університет, кандидат с.-г. наук, доцент	5	2	0
Налія Анатоліївна Меженська старший науковий співробітник ДНДІЛДВСЕ	6	2	0

Рисунок 3.7 – Вигляд сторінки веб-системи після застосування фільтру за підгалуззями

3.4. Рейтингування науковців

Для того щоб рейтингувати науковців в веб-системі наявні два варіанти функції сортування. Одна з них немає можливість швидко і просто відсортувати науковців в порядку спадання за одним з трьох показників цитування. Вигляд випадаючого списку даної функції в інтерфейсі системи показано на рисунку 3.8.

The screenshot shows the 'Sort by' dropdown menu open, with 'h-index' selected. The search results list one scientist:

Scientist Name	Citations	h-index	i10-index
Володимир Ятченко Інститут сцинтиляційних матеріалів НАН України, Institute for Scintillation Materials	43	4	1

Рисунок 3.8 – Вибір параметра сортування науковців з у відповідності до одного з трьох показників цитування

Після вибору за яким показником цитування здійснюватиметься сортування у відповідності до його значень серед науковців. Після здійснення сортування за і10-індексом сторінка веб-системи набуває вигляду показаного на рисунку 3.9.

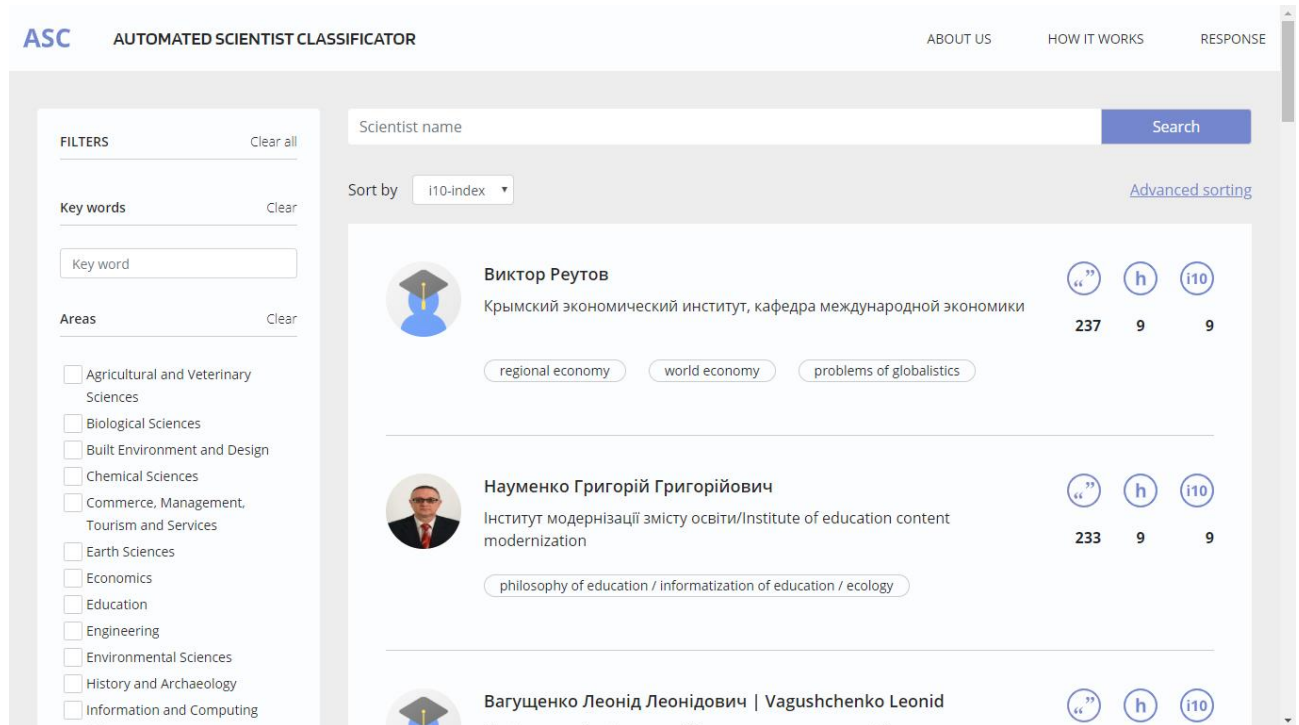


Рисунок 3.9 – Вигляд сторінки веб-системи після виконання сортування за і10-індексом

Іншим варіантом сортування є розширене сортування яке розуміє під собою задання за яким показником цитування буде здійснюватися сортування в першу чергу, в другу чергу, і в третю чергу. Крім того користувач може обрати за показниками з якого часового періоду буде здійснюватися сортування чи за показниками за весь час чи лише за показниками за останні 5 років.

Задання параметрів розширеного сортування здійснюємо з допомогою елементів інтерфейсу показних на рисунку 3.10

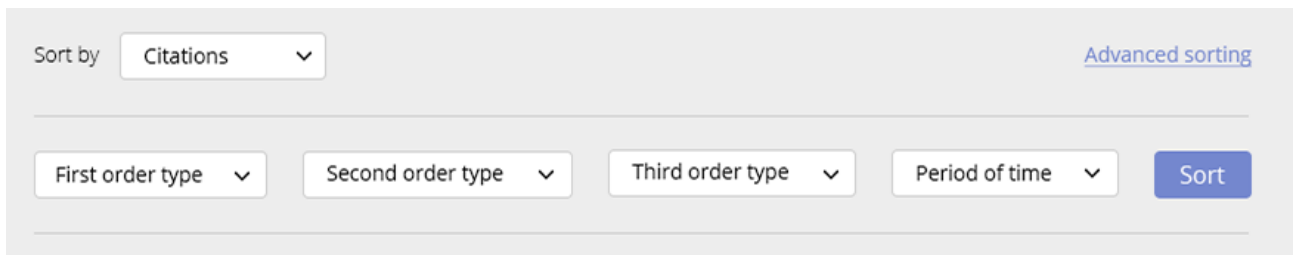


Рисунок 3.10 – Елементи інтересу через які задаються параметри розширеного сортування

Після задання параметрів я пов'язаних з показниками цитування можна одразу натиснути кнопку «Sort» в результаті чого буде здійснено сортування за показниками цитування за весь час. Результат такого сортування показано на рисунку 3.11.

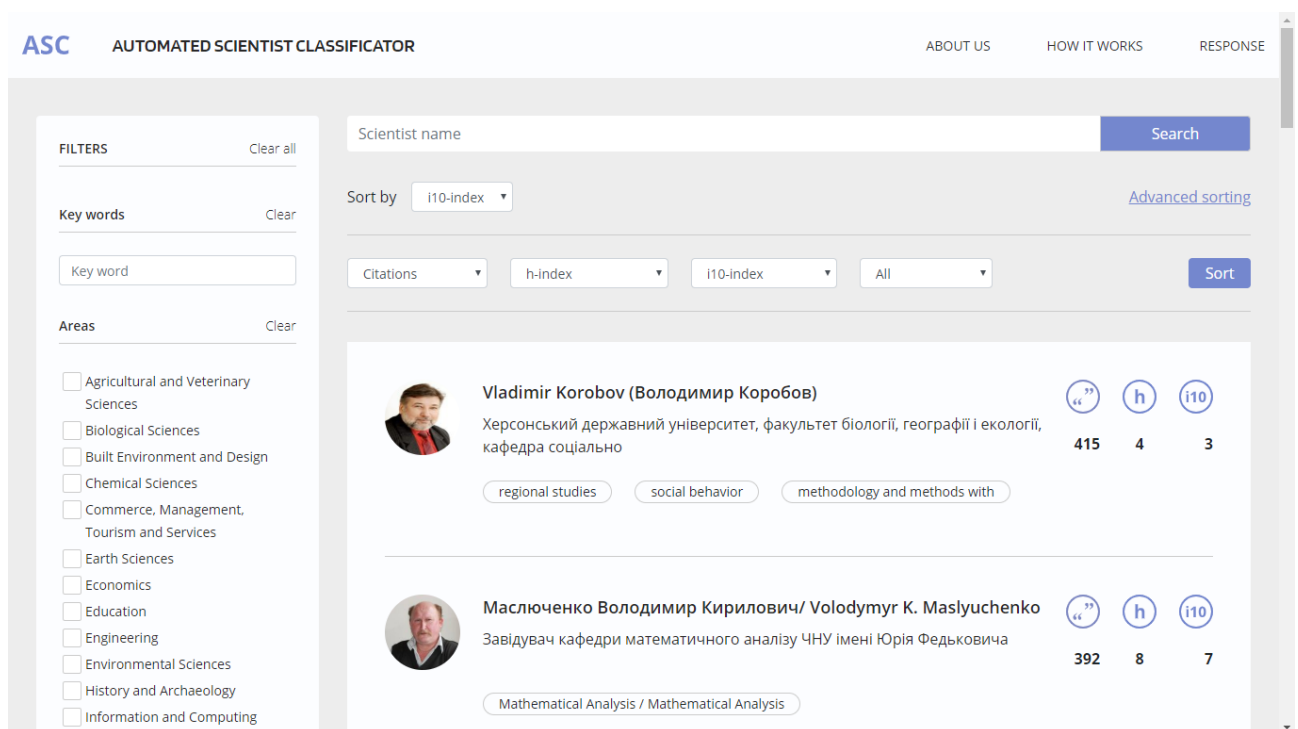


Рисунок 3.11 – Вигляд сторінки веб-системи після здійснення розширеного сортування за показниками цитування за усі роки

Якщо ціллю користувача є отримання інформації про науковців які мають найбільші показники цитування за останні 5 років тоді крім основних трьох параметрів за показниками цитування потрібно ще задати четвертий параметр

обравши пункт «Last 5 years». Результат сортування науковців за показниками останніх 5 років показано на рисунку 3.12.

The screenshot shows the ASC (Automated Scientist Classifier) web interface. On the left, there are filters for 'Key words' and 'Areas'. The main area displays search results for scientists, sorted by 'i10-index' for the 'Last 5 years'. Two scientists are listed:

Scientist Name	Institution	Citations	h-index	i10-index
Вадим Володимирович Тютюник / Вадим Владимирович Тютюник / Vadim Tiutiunyk	National University of Civil Protection of Ukraine / Національний університет цивільного захисту	284	9	8
Сергій Головащенко, С. Головащенко, Головащенко С. І., Sergii Golovashchenko, Serhij Hołowaszczenko	National University of Kyiv-Mohyla Academy / Національний університет "Киево-Могилянська академія"	425	9	9

Рисунок 3.11 – Вигляд сторінки веб-системи після здійснення розширеного сортування за показниками цитування останніх 5 років

3.5. Дослідження розподілів науковців

Значення цитування науковця є важливим показником для отримання інформації про інтенсивність та вагомість його наукової діяльності. Зрозуміло що якщо у науковця високі показники цитування то це свідчить про його активну наукову діяльність, а також свідчить про те що його наукова діяльність є важливою для наукової спільноти і не лише зважаючи на те що багато людей на неї посилаються.

Цікавим фактом те що показники цитування певного набору науковців відповідає закону Ципфа. Початково закон Ципфа знайшов своє застосування до дослідження частоти слів у тій чи іншій природній мові. Даний закон говорить про те що якщо впорядкувати слова достатньо великого тексту за частотою їхнього використання, то частота n -го слова виявиться приблизно обернено пропорційною його порядковому номеру n (рангу) [49].

В результаті такої залежності дані розташовуються у вигляді степеневого розподілу. Після відкриття закону Ципфа він знайшов своє підтвердження не лише у природній мові але й у багатьох інших речах таких як частоті прізвищ, населені міст, числі проданих книг і в багатьох інших. Відповідно закон Ципфа знайшов своє підтвердження і у показниках цитування науковців.

Формулу закону Ципфа можна представити наступним чином [50]:

$$f(r) = \frac{a}{r^\gamma} \quad (3.1)$$

У своїй спробі вивести закон Ципфа з теорії інформації Мандельброт створив її більш узагальнену версію яка виглядає наступним чином [51, 52]:

$$f(r) = \frac{P_1}{(P_2 + r)^{P_3}} \quad (3.2)$$

де P_1, P_2, P_3 – це константи.

3.5.1. Розподіли за цитованістю

Візьмемо дані значень цитування усіх науковців які є в базі даних впорядкуємо їх в порядку спадання і далі при побудові графіка за значеннями цитування і їхніми рангами отримаємо результат який показано на рисунку 3.12.

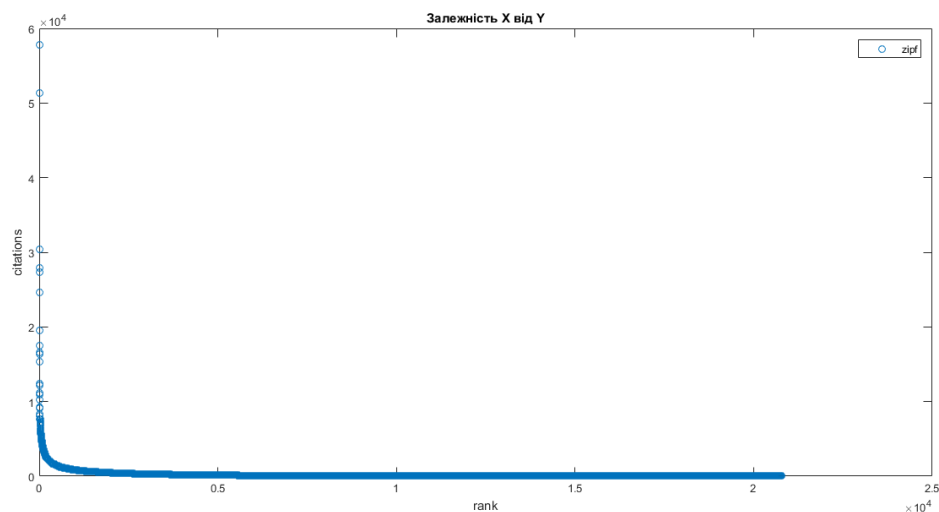
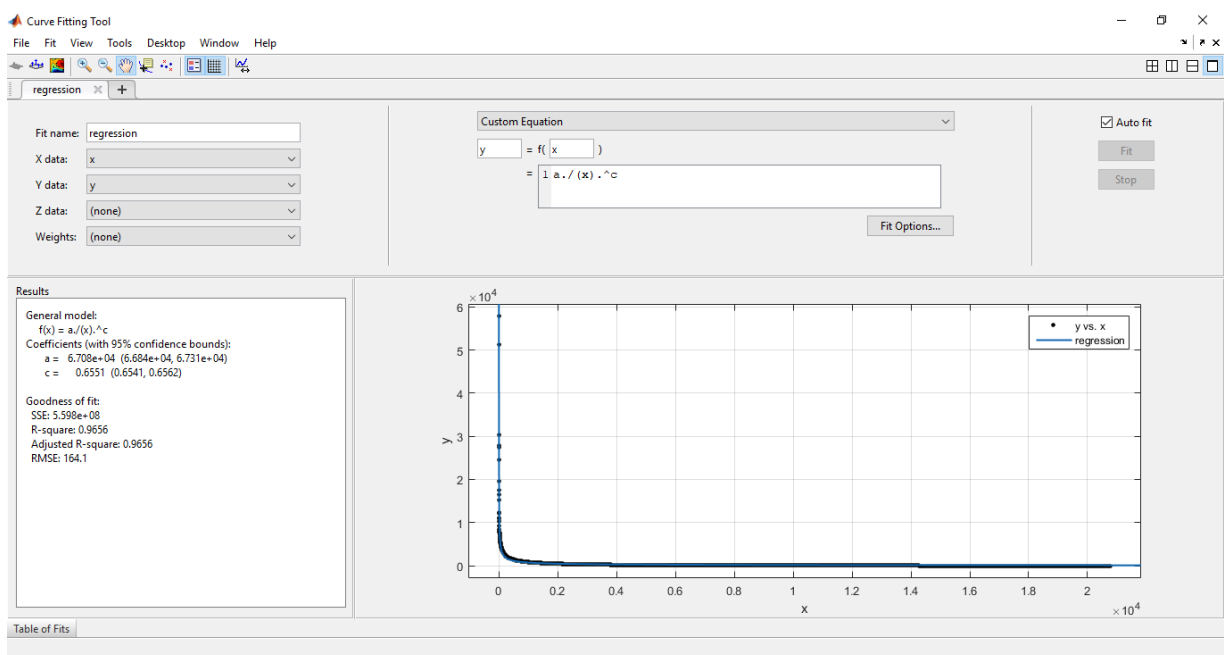


Рисунок 3.12 – Графік залежності значення цитування від його рангу

Як видно з рисунку 3.12 відношення значень цитування до їх рангів науковців які є у базі системи також являє собою степеневий розподіл і відповідно відповідають закону Ципфа.

Ми можемо віднайти залежність яка найкращим чином наявні у нас значення цитування науковців. Для того щоб віднайти залежність яка найкращим чином описує наявні у нас дані скористаємося регресійним аналізом. Як відомо регресійний аналіз розуміє під собою визначення критеріїв залежності $y = f(x)$ при яких мінімізується нев'язка RMSE. В якості залежності за якою будуть визначатися коефіцієнти візьмемо формулу закону Ципфа (3.1). В даному випадку ми маємо багато варіантів як підійти до самого регресійного аналізу і оптимальних коефіцієнтів. Однак зважаючи на наявність зараз програмних засобів які дозволяють виконати дане завдання без затрати значного часу на здійснення всіх обчислень ми скористаємося інструментом з оптимізації «Curve Fitting Tool» програмного пакету Matlab. Задавши усі необхідні параметри в даному інструменті такі як початкові дані і функції за якою буде здійснюватися пошук ми отримаємо результат показаний на рисунку 3.13.



Рисунку 3.13 – Вигляд інструменту «Curve Fitting Tool»

В даному вікні містяться графік який відображає розподіл цитування а також регресію яка найкращим чином описує даний розподіл, елементи для задання даних для регресійного аналізу, та блок результатів в якому відображаються обраховані коефіцієнти та середня квадратична нев'язка. Графік на якому відображені дані цитування та регресія показано на рисунку 3.14.

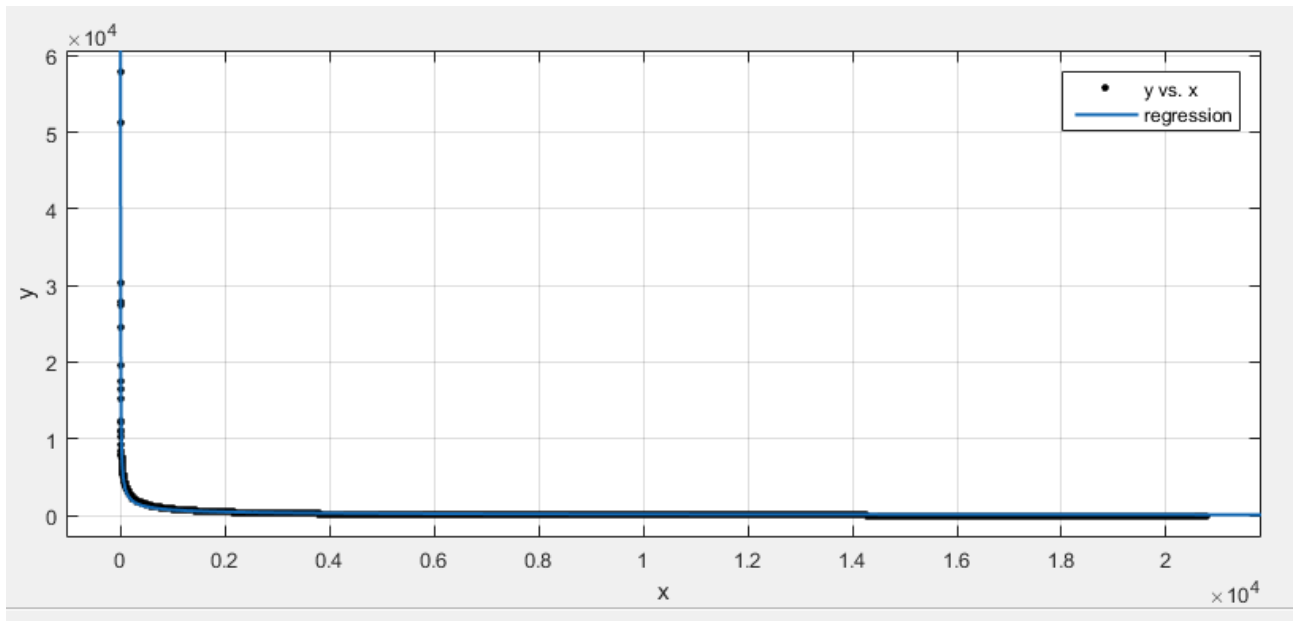


Рисунок 3.14 – Результат регресійного аналізу за всіма значеннями цитування

В результаті даного регресійного аналізу ми отримали наступні значення коефіцієнтів регресійної моделі $f(x) = a/(x)^c$ та RMSE:

$$a = 6.708e+04,$$

$$c = 0.6551,$$

$$RMSE = 164.1$$

3.5.2. Розподіли за галузями

Розгляньмо розподіли за значеннями цитування в окремих галузях.

Виберемо для цього наступні галузі науки: Chemical Sciences, Built Environment and Design та Education.

На рисунку 3.15 показано графік даних цитування за галуззю «Chemical Sciences» та їхню регресію.

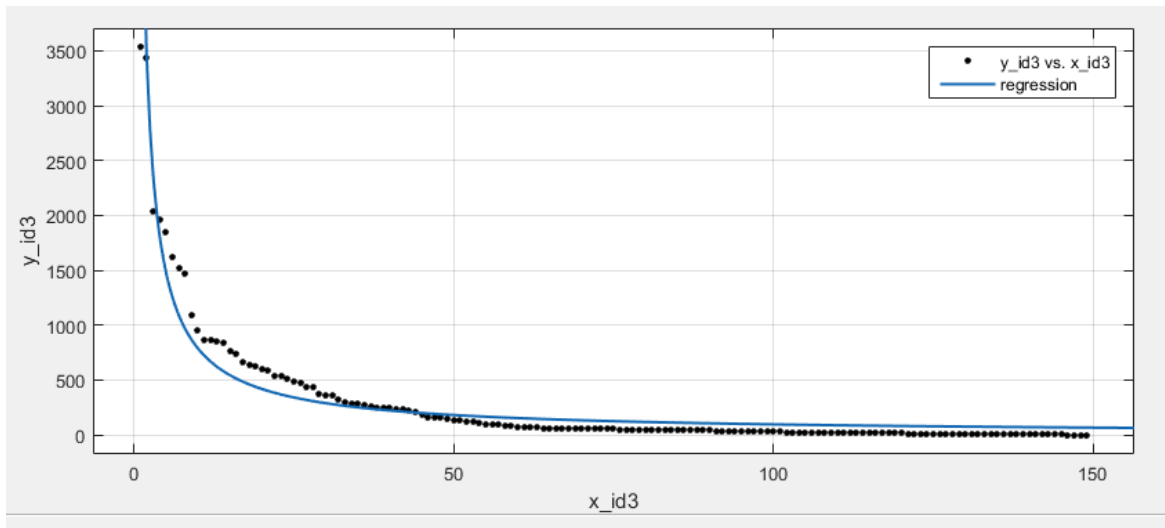


Рисунок 3.15 – Результат регресійного аналізу за всіма значеннями цитування галузі «Chemical Sciences»

В результаті даного регресійного аналізу ми отримали наступні значення коефіцієнтів регресійної моделі $f(x) = a/(x)^c$ та RMSE:

$$a = 6474,$$

$$c = 0.9143,$$

$$\text{RMSE} = 28.79$$

На рисунку 3.16 показано графік даних цитування за галуззю «Built Environment and Design» та їхню регресію.

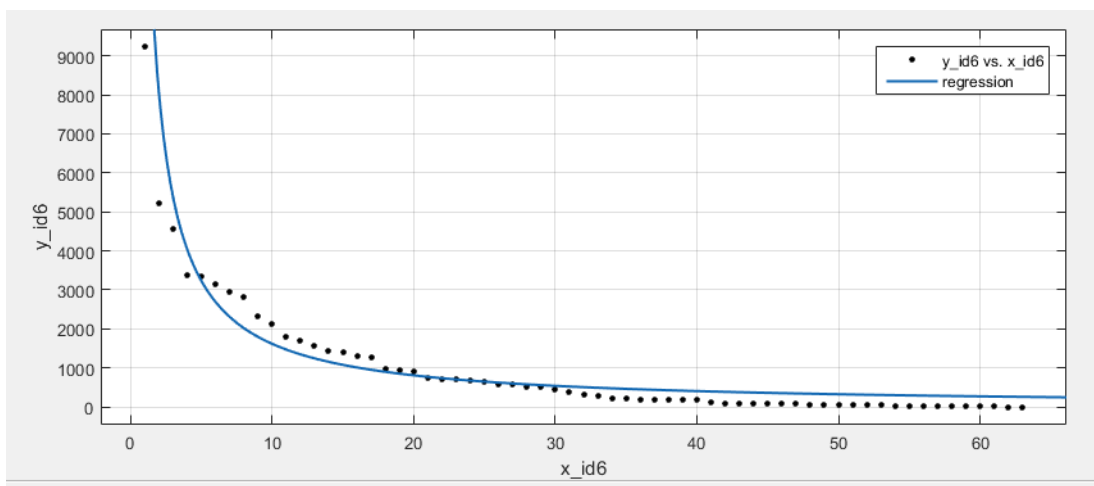


Рисунок 3.16 – Результат регресійного аналізу за всіма значеннями цитування галузі «Built Environment and Design»

В результаті даного регресійного аналізу ми отримали наступні значення коефіцієнтів регресійної моделі $f(x) = a/(x)^c$ та RMSE:

$$a = 1.616e+04,$$

$$c = 1,$$

$$RMSE = 333.7$$

На рисунку 3.17 показано графік даних цитування за галуззю «Education» та їхню регресію.

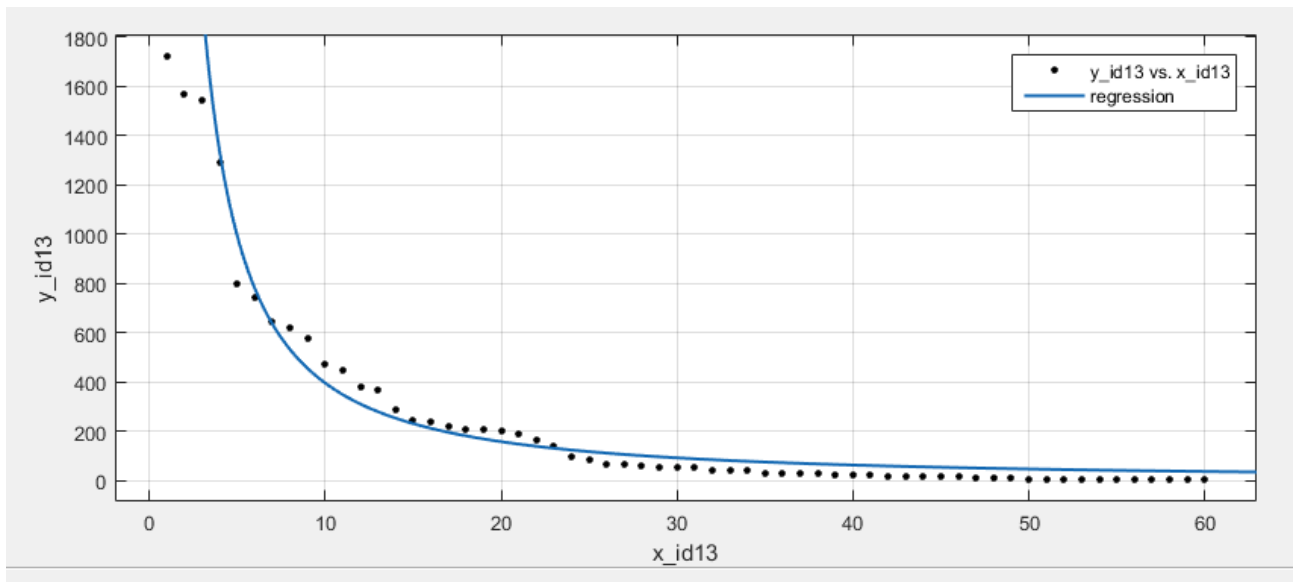


Рисунок 3.17 – Результат регресійного аналізу за всіма значеннями цитування галузі «Built Environment and Design»

В результаті даного регресійного аналізу ми отримали наступні значення коефіцієнтів регресійної моделі $f(x) = a/(x)^c$ та RMSE:

$$a = 8403,$$

$$c = 1.328,$$

$$RMSE = 65.2$$

Зважаючи на вище наведені графіки та показники можна зауважити що дані цитування відповідають закону Ципфа навіть якщо брати до уваги

невеликий набір значень наприклад як в даному випадку значень цитувань окремої галузі.

3.5.3. Розподіли за Хіршем

Розглянемо розподіл не безпосередньо значень цитування а значень показника який їх характеризує, а саме значення індексу Хірша. Подувавши для початку графік за значеннями індексу Хірша усіх науковці що містяться в базі даних отримаємо результат показаний на рисунку 3.18.

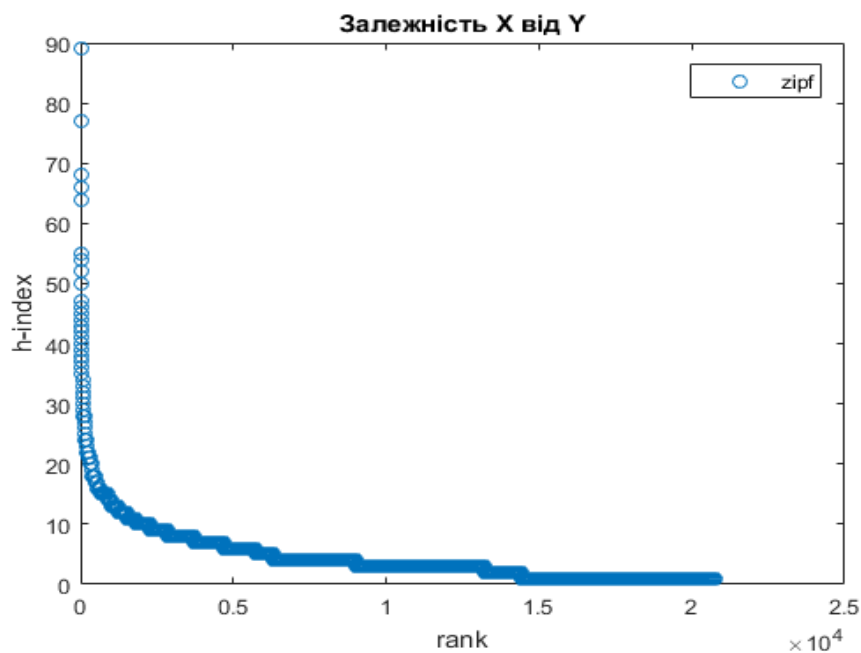


Рисунок 3.18 – Розподіл значень індексу Хірша усіх наявних науковців

Зважаючи на те як видно із зображення що розподіл значень індексу Хірша серед усіх науковців відповідає законі Ципфа то здійснимо регресійний аналіз за ними. На рисунку 3.19 показано графік даних індексу Хірша та їхню регресію.

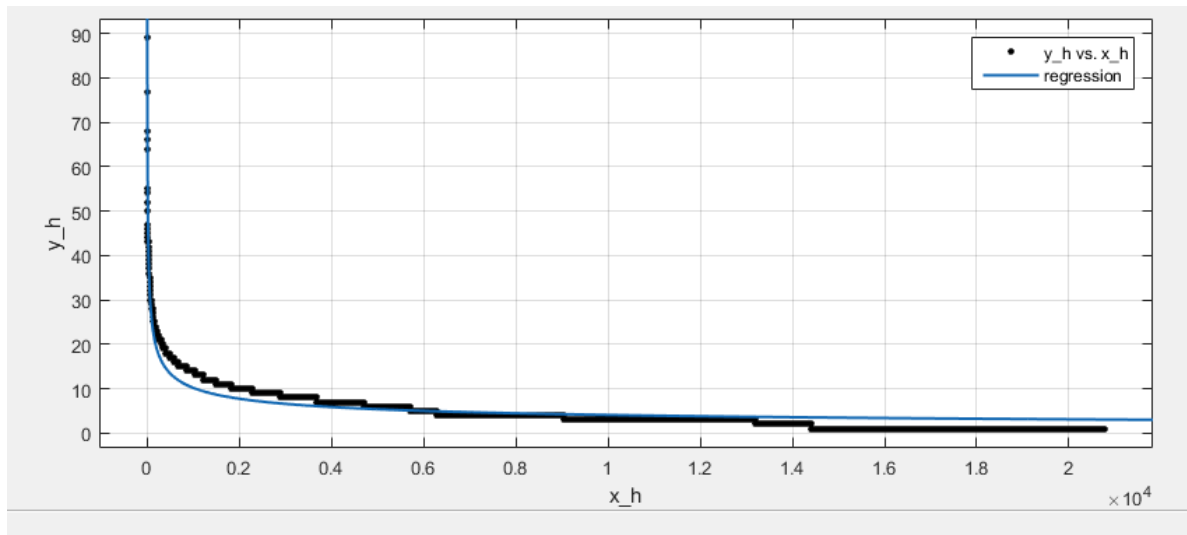


Рисунок 3.19 – Результат регресійного аналізу за всіма значеннями індексу Хірша

В результаті даного регресійного аналізу ми отримали наступні значення коефіцієнтів регресійної моделі $f(x) = a/(x)^c$ та RMSE:

$$a = 160.5,$$

$$c = 0.3992,$$

$$\text{RMSE} = 1.979$$

3.6. Висновки до розділу

В даному розділі було описано проведення тестування розробленої системи. Тестування було проведено у відповідності до головних функцій які має виконувати система, а саме пошуку науковця, пошуку науковців за спеціальностями, пошук науковців за підгалузями науки та рейтингуванням науковців.

Кожну з цих функцій системи було перевірено у розробленому веб-інтерфейсі користувача на предмет коректної роботи та відсутності яких небуть помилок. Тестування показали що система працює коректно і під час роботи з нею не виникає жодних помилок. Також в ході тестування функцій системи і самого її інтерфейсу було підтверджено зручність розташування елементів що дозволить навіть незнайомому з інтерфейсом користувачу легко та зручно користуватися функціями які надає система.

В даному розділі було проведено експериментальні дослідження над показниками цитування за у відповідності до закону Ципфа. Закон Ципфа відображає доволі цікаву залежність між значеннями цитування та їхніми рангами що дозволяє впорядкувати науковців у відповідності до цих значень. Враховуючи це провівши додаткові дослідження можна знайти застосування розподілу цитувань за законом Ципфа у інформаційній системі ієрархічної категоризації науковців.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1. Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки інформаційної системи ієрархічної категоризації науковців за відкритими даними з Google Scholar, яка підвище точність та продуктивність існуючого підходу до категоризації науковців.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів Вінницького національного технічного університету, кафедри комп'ютерних систем управління: Штовба С. Д., Ковалюк О. О., Ковтун В. В. За допомогою таблиці 4.1 за п'ятибальною шкалою використовуючи 12 критеріїв оцінки комерційного потенціалу розробки експерти надали свої оцінки.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження табл. 4.1

5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання	Необхідно отримання великої кількості дозвільних документів на виробництво та	Процедура отримання дозвільних документів для	Необхідно тільки повідомлення відповідним органам про	Відсутні будь-які регламентні обмеження на виробництво та реалізацію

	великої кількості дозвільних документів на виробництво та реалізацію продукту	реалізацію продукту, що вимагає значних коштів та часу	виробництва та реалізації продукту вимагає незначних коштів та часу	виробництво та реалізацію продукту	продукту
--	---	--	---	------------------------------------	----------

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 4.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Штовба С.Д	Ковалюк О.О.	Ковтун В.В.
	Бали, виставлені експертами:		
1	3	2	2
2	3	3	3
3	3	3	2
4	4	3	3
5	4	4	4
6	2	1	1
7	3	3	4
8	3	4	4
9	2	2	3
10	4	4	4
11	4	4	4
12	4	4	4
Сума балів	СБ ₁ =39	СБ ₂ =37	СБ ₃ =38
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{39+37+38}{3} = 38$		

Середньоарифметична сума балів, розрахована на основі висновків експертів склала 38, що згідно таблиці 4.2 вважається, що рівень комерційного потенціалу розробки є вище середнього.

Розробка буде реалізована в університеті та буде виставлена на його сайті. Ймовірні користувачі: науковці, студенти для пошуку інших науковців зі схожими науковими інтересами, для пошуку рецензентів робіт, для пошуку наукових керівників, для пошуку фахівців певної галузі.

Проведемо порівняння розробки, яка розробляється з аналогом, який існує на ринку. В якості аналога для розробки було обрано бібліометрику української науки. Основними недоліками аналога є суб'єктивність при здійсненні категоризації. Також до недоліків можна віднести обмеженість у системі наук та однозначна відповідність між науковцем та галуззю науки

У розробці дана проблема вирішується шляхом застосування методів штучного інтелекту. Також система випереджає аналог за такими параметрами як – швидкодія, кількість людських ресурсів.

В таблиці 4.4 наведені основні техніко-економічні показники аналога і нової розробки.

Таблиця 4.4 - Основні технічні показники аналога і нової розробки

Показники	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Кількість галузей науки	12	22	1.83
Швидкодія	1	10	10
Витрати на виконання роботи	1	10	10
Кількість підгалузей	180	154	0.85
Нерівномірність розподілу	1	2	2

Виходячи із даних таблиці 4.4 можна зробити висновок, що у зв'язку з швидким розвитком науки і техніки все більше і більше набуває популярності

автоматизація різноманітних процесів. Саме цей показник відсутній у системі конкурента – усі дії пов’язані з категоризацією виконуються людиною, тобто суб’єктивно, з певними затратами часу та більш затратно у порівнянні з автоматичною системою.

4.2. Прогнозування витрат на виконання науково-дослідної роботи

Проведемо прогнозування витрат на виконання робіт у три етапи:

1-й етап: Розрахунок витрат, які безпосередньо стосуються виконавців даного розділу НДР.

2-й етап: Розрахунок загальних витрат на виконання НДР;

3-й етап: Прогнозування загальних витрат на виконання та впровадження НДР.

Проведемо перший етап прогнозування.

1. Основна заробітна плата кожного із дослідників Z_0 , якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$Z_0 = \frac{M}{T_p} * t \text{ (грн)} \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_p – число робочих днів в місяці; приблизно $T_p \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

Дану розробку буде проводити інженер-програміст, величина окладу буде становити 6500 грн. на місяць. Кількість робочих днів у місяці складає 22, а кількість робочих днів дослідника складає 55. Зведемо сумарні розрахунки до таблиця 4.5.

Таблиця 4.5 – Заробітна плата дослідника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату грн.
Керівник проекту	12000	545.5	5	2727
Інженер-програміст	6500	295.5	55	16250
Всього				18977

2. Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата Z_d всіх розробників та робітників, які приймали участь в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників. На даному підприємстві додаткова заробітна плата начисляється в розмірі 10% від основної заробітної плати.

$$Z_d = 0,11 * 18977 = 2087,5 \text{ (грн)} \quad (4.2)$$

3. Нарахування на заробітну плату $H_{ЗП}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою (4.3):

$$H_{ЗП} = (Z_o + Z_d) * \frac{\beta}{100} \text{ (грн)} \quad (4.3)$$

де Z_o – основна заробітна плата розробників, грн.;

Z_d – додаткова заробітна плата всіх розробників та робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді:

$$H_{3П} = (18977 + 2087,5) * \frac{22}{100} = 4634,3 \text{ (грн)}$$

4. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

$$A = \frac{Ц \cdot T}{T_{кор} \cdot 12} \text{ [грн]}, \quad (4.4)$$

де Ц – балансова вартість даного виду обладнання (приміщень), грн.;

$T_{кор}$ – час користування;

T – термін використання обладнання (приміщень), цілі місяці.

Згідно пункту 137.3.3 Податкового кодекса амортизація нараховується на основні засоби вартістю понад 2500 грн. В нашому випадку для написання магістерської роботи використовувався персональний комп'ютер вартістю 25000 грн.

$$A = \frac{25000 \cdot 2}{2 \cdot 12} = 2083,3$$

5. Норма витрат матеріалу – це плановий показник, який визначає максимально допустимі затрати відповідних ресурсів на виробництво одиниці продукції в умовах певного рівня техніки і організації виробництва.

Витрати на матеріали М, що були використані під час виконання даного етапу роботи, розраховуються по кожному виду матеріалів за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i - \sum_1^n B_i \cdot Ц_b \text{ грн.}, \quad (4.5)$$

де H_i – витрати матеріалу i -го найменування, кг;

$Ц_i$ – вартість матеріалу i -го найменування, грн./кг.;

K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$;

B_i – маса відходів матеріалу i -го найменування, кг;

$Ц_b$ – ціна відходів матеріалу i -го найменування, грн/кг;

n – кількість видів матеріалів.

Інформацію про використанні матеріали подамо у вигляді табл. 4.6.

Таблиця 4.6– Матеріали, що використанні на розробку

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Папір	75	1	75
Ручка	12	1	12
Флешка	120	1	120
CD-диск	15	1	15
З врахуванням коефіцієнта транспортування			244.2

6. До статті «Паливо та енергія на технологічні цілі» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються у процесі виробництва продукції. У даному випадку будемо враховувати лише витрати на електроенергію, яку споживає комп'ютер:

$$V_e = V \cdot P \cdot \Phi \cdot K_n, \quad (4.6)$$

де V – вартість 1 кВт енергії, грн. $V = 8,44$ грн/кВт*год;

P – установлена потужність обладнання, кВт. $P = 500$ Вт або $P = 0,5$ кВт;

Φ – фактична кількість годин роботи обладнання, год. $\Phi = 200$ год.;

K_n – коефіцієнт використання потужності, $K_n = 0,65$.

$$V_e = 8,44 \cdot 0,5 \cdot 200 \cdot 0,65 = 548,6(\text{грн}).$$

Інші витрати B_{in} охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Інші витрати B_{in} можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$B_{in} = (1..3) \cdot (3 + 3_p). \quad (4.7)$$

$$B_{in} = 1 \cdot 18977 = 18977 (\text{грн.})$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР

$$B = 18977 + 2087,5 + 4634,3 + 2083,3 + 244,2 + 548,6 + 18977 = 47552,4 (\text{грн.})$$

Загальна вартість всієї МКНР визначається за формулою:

$$B_{заг} = \frac{B}{\alpha} \quad (4.8)$$

$$B_{заг} = \frac{47552,4}{1} = 47552,4 (\text{грн.})$$

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{B}{\beta}, \quad (4.9)$$

де β – коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт $\beta = 0,9$.

Звідси:

$$ЗВ = \frac{47552,4}{0,9} = 52836 (\text{грн.}).$$

4.3. Оцінка внеску НДР

Для цього спочатку потрібно визначити коефіцієнт наукової значимості отриманих результатів НДР, а потім підрахувати внесок в досягнення цих результатів.

Коефіцієнт наукової значимості результатів проведеної науково-дослідної роботи $K_{ЗН}$ можна підрахувати за формулою:

$$K_{ЗН} = \frac{\sum_1^3 b_i * d_i}{\sum_1^3 b_{max} * d_i} \quad (4.10)$$

де b_i – значимість отриманих результатів: b_1 – ступінь наукової новизни, b_2 – рівень теоретичної обґрунтованості, b_3 – ступінь експериментальної перевірки результатів. Бальна оцінка отриманих результатів наведена в таблиці 4.8. Максимальне значення отриманих за кожною характеристикою результатів можна прийняти в межах 7...10 балів;

d_i – питома вага кожної характеристики, значення якої наведені в таблиці 4.8;

3 – кількість характеристик, за якими була зроблена оцінка результатів науково-дослідної роботи.

Таблиця 4.8 – Показники для оцінювання наукової значимості результатів виконаної НДР

Характеристики	Питома вага характеристики d_i	Бальна оцінка характеристики		
		Ступінь новизни b_1	Рівень теоретичної обґрунтованості b_2	Ступінь експериментальної перевірки результатів b_3
		1	2-5	7-10
b_1	0,500	Часткове удосконалення виробів, технологій, матеріалів, програмного продукту тощо	Суттєве удосконалення виробів, технологій, матеріалів, програмного продукту тощо	Нові напрямки в розробці виробів, технологій, матеріалів, програмного продукту тощо. Створення принципово нової

				техніки
b_2	0,333	Позитивне рішення на основі зроблених узагальнень	Установлення залежності, які використовувались в інших випадках	Відкриття нових шляхів рішення задачі
b_3	0,167	Експериментальна перевірка не робилась	Результати перевірялись на невеликій кількості даних	Результати перевірені на великій кількості даних

В ході виконання даної магістерської кваліфікаційної роботи було розроблено інформаційну систему ієрархічної категоризації науковців за відкритими даними з Google Scholar, що підвищить точність та продуктивність існуючого підходу до категоризації науковців.

Тоді згідно таблиці 4.8 $b_1 = 4$. Отримані результати підтвердили залежності та характеристики, отримані раніше для окремих випадків рішення даної задачі, тому $b_2 = 7$. Оскільки результати перевірялись на отриманих попередньо експериментальних даних, приймаємо $b_3 = 5$. Тоді коефіцієнт наукової значимості результатів буде наступним

$$K_{3H} = \frac{4 \cdot 0,5 + 7 \cdot 0,333 + 5 \cdot 0,167}{10 \cdot 0,5 + 10 \cdot 0,333 + 10 \cdot 0,167} = 0,52.$$

Зважаючи на малий час, відведений на проведення дослідження, отримане значення коефіцієнта наукової значимості можна вважати цілком задовільним.

Внесок дипломника в досягнення отриманих результатів НДР можна розрахувати за формулою:

$$V = \frac{k_{\text{ТВИ}} * Z_i}{\sum_i^n k_{\text{ТВИ}} * Z_i} \quad (4.11)$$

де $k_{\text{ТВИ}}$ – коефіцієнт творчої участі кожного виконавця НДР, який оцінюється таким чином: проведення досліджень – 3 бали, робоче проектування – 1,5 бали, освоєння – 1,0 балів. якщо виконавець приймав участь в декількох видах робіт, то береться сума відповідних балів;

Z_i – заробітна плата кожного виконавця НДР, грн.;

n – кількість всіх виконавців науково-дослідної роботи: наукові керівники, відповідальні виконавці, спеціалісти, наукові співробітники, робітники різних професій тощо.

$$V = \frac{3 * 12000}{3 * 12000 + 6500 * 1,5} = 0,77$$

4.4. Висновок

В даному розділі було оцінено комерційний потенціал розробки інформаційної системи ієрархічної категоризації науковців за відкритими даними з Google Scholar, яка підвищить точність та продуктивність існуючого підходу до категоризації науковців. Комерційний потенціал розробки виявився на вище середньому рівні.

Прогнозування витрат на виконання науково-дослідної роботи по кожній з статей витрат складе 47552,4 грн. Загальна ж величина витрат на виконання та впровадження результатів даної НДР буде складати 52836 грн.

Отримане значення коефіцієнта наукової значимості ($K_{\text{ЗН}} = 0,52$) можна вважати цілком задовільним, оскільки час відведений на проведення дослідження незначний. Особистий внесок дипломника в наукове дослідження становить 0,77, що являється нормальним значенням.

ВИСНОВКИ

В роботі було проведено аналіз існуючих інформаційно-пошукових систем, було досліджено те якими вони бувають і які сервіси надають користувач. Було здійснено варіативний огляд сучасних технологій з допомогою яких можна реалізувати веб-систему ієрархічної категоризації науковців. Серед технологій які розглядалися було вирішено реалізувати систему на базі фреймворку Bootstrap з огляду на підтримку адаптивності та простоту в освоєнні.

Для того щоб визначити яким чином працюватиме система було визначено її головні функції. Після визначення функцій системи використовуючи UML моделювання було створено функціональну схему інші схеми які усі разом являють собою архітектуру системи. У відповідності до функцій системи і принципу їхньої майбутньої роботи було розроблено веб-інтерфейс користувача.

Розроблений інтерфейс веб-системи містить усі необхідні елементи які дозволяють використовувати такі функції системи як пошук науковця, пошук науковця за галуззю, рейтингування науковців. Проведенні тестування показали коректну роботу усіх складових елементів системи та її функцій. Значення загальних показників цитування, показників цитування в окремих галуззях та загальних індексів Хірша було досліджено на відповідність закону Ципфа. В ході дослідження було підтверджено що дані показники можна описати з допомогою закону Ципфа. Враховуючи те яку інформацію яку можна отримати при описі даною залежністю можна сказати що при подальших дослідженнях на базі понять закону Ципфа можна буде розширити функціонал системи ієрархічної категоризації науковців.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ІНФОРМАЦІЙНО-ПОШУКОВІ СИСТЕМИ [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2014-2019 Інститут енциклопедичних досліджень НАН України – Режим доступу: http://esu.com.ua/search_articles.php?id=12483– Назва з екрана.
2. Information retrieval [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: https://en.wikipedia.org/wiki/Information_retrieval– Назва з екрана.
3. Information Retrieval Systems [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems>– Назва з екрана.
4. Інформаційно-пошукові системи [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://sites.google.com/site/socialnafnformatika/home/informacijno-posukovi-sistemi>– Назва з екрана.
5. Поняття і класифікація інформаційно-пошукових систем [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://library.if.ua/book/100/6867.html>– Назва з екрана.
6. Gitte Lindgaard, Gary Fernandes, Cathy Dudek & J. Brown (2006) Attention web designers: You have 50 milliseconds to make a good first impression!, Behaviour & Information Technology, 25:2, 115-126, DOI: 10.1080/01449290500330448
7. 15 Website & UX Statistics of 2018-2019 [INFOGRAPHIC] [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : UKWebHostReview.com 2019 – Режим доступу: <https://www.ukwebhostreview.com/website-and-ux-statistics/>– Назва з екрана.
8. Використовність [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Wikipedia 2019 – Режим доступу: <https://uk.wikipedia.org/wiki/Використовність> – Назва з екрана.
9. ISO 9241-210:2019 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>
10. Что такое юзабилити или 7 основных правил юзабилити сайта [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Medium 2019 – Режим доступу: https://medium.com/@rinat_sh/@rinat_sh/что-такое-юзабилити-или-7-основных-правил-юзабилити-сайта-3894d9ba1437 – Назва з екрана.

11. Досвід користування [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Wikipedia 2019 – Режим доступу: https://uk.wikipedia.org/wiki/Досвід_користування – Назва з екрана.
12. An Introduction to Usability and User Experience in Web Design [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : A2 HOSTING 2019 – Режим доступу: <https://www.a2hosting.com/blog/usability-and-user-experience/> – Назва з екрана.
13. Usability 101: Introduction to Usability [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Nielsen Norman Group 2019 – Режим доступу: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> – Назва з екрана.
14. 5 Key Principles Of Good Website Usability [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Crazy Egg, Inc 2019 – Режим доступу: <https://www.crazyegg.com/blog/principles-website-usability/> – Назва з екрана.
15. Share of mobile phone website traffic worldwide 2018 [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Statista 2018 – Режим доступу: <https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/> – Назва з екрана.
16. DIGITAL IN 2018: WORLD'S INTERNET USERS PASS THE 4 BILLION MARK [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2008–2019 We Are Social Ltd. – Режим доступу: <https://wearesocial.com/blog/2018/01/global-digital-report-2018> – Назва з екрана.
17. Mobie vs. Desktop Usage (Latest 2019 Data) [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics> – Назва з екрана.
18. Should You Have a Professional Website Design? Absolutely, Here's Why. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://www.lyfemarketing.com/blog/professional-website-design/> – Назва з екрана.
19. Handbook of Human-Computer Interaction – M.G. Helander, T.K. Landauer, P.V. Prabhu Elsevier, 18 серп. 1997 р. - 1582 стор.
20. Effective Visual Communication for Graphical User Interfaces [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: https://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html – Назва з екрана.
21. Usability and User Experience Design Stats [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://www.tytonmedia.com/blog/51-insane-web-design-statistics-2016/> – Назва з екрана.

22. 17 Reasons Why Your Website Should Have a Clean and Simple Design [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://neilpatel.com/blog/website-clean-simple-design/> – Назва з екрана.
23. Predicting Users' First Impressions of Website Aesthetics With a Quantification of Perceived Visual Complexity and Colorfulness [Електронний ресурс] – Електронні дані. – : 2019 – Режим доступу: <http://www.eecs.harvard.edu/~kgajos/papers/2013/reinecke13aesthetics.pdf> – Назва з екрана.
24. 8 Guidelines for Exceptional Web Design, Usability, and User Experience [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : HubSpot, Inc 2019 – Режим доступу: <https://blog.hubspot.com/blog/tabid/6307/bid/30557/6-guidelines-for-exceptional-website-design-and-usability.aspx> – Назва з екрана.
25. Features of the Perfect Responsive Web Design [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://designmodo.com/prefect-responsive-web-design/> – Назва з екрана.
26. What Is Responsive Web Design? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://designadvisor.net/blog/what-is-responsive-web-design/> – Назва з екрана.
27. Resolution dependent layout [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <http://www.themaninblue.com/writing/perspective/2004/09/21/> Назва з екрана.
28. Responsive Web Design [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2010 – Режим доступу: <https://alistapart.com/article/responsive-web-design/> – Назва з екрана.
29. Responsive Web Design [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : SitePoint Pty. Ltd. 2012 – Режим доступу: <https://www.sitepoint.com/responsive-web-design/> – Назва з екрана.
30. A Beginner's Guide to CSS Front End Frameworks [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://blog.zipboard.co/a-beginners-guide-to-css-front-end-frameworks-8045a499456b> – Назва з екрана.
31. What are Frameworks? 22 Best Responsive CSS Frameworks for Web Design [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html> – Назва з екрана.
32. Why Libraries are better than Frameworks [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <http://tom.lokhorst.eu/2010/09/why-libraries-are-better-than-frameworks> – Назва з екрана.

33. Top 10 Front-End Frameworks of 2018 [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : KeyCDN 2019 – Режим доступу: <https://www.keycdn.com/blog/front-end-frameworks> – Назва з екрана.
34. BOOTSTRAP HISTORY [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Education Ecosystem 2019 – Режим доступу: <https://www.education-ecosystem.com/guides/programming/bootstrap/history>– Назва з екрана.
35. Bootstrap [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://getbootstrap.com/>– Назва з екрана.
36. Foundation vs. Bootstrap: Which front end framework to use? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://medium.com/@davegenge/bootstrap-vs-foundation-which-front-end-framework-to-use-e85319258b88> – Назва з екрана.
37. The 5 Most Popular Front-end Frameworks Compared [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : SitePoint Pty. Ltd. 2019 – Режим доступу: <https://www.sitepoint.com/most-popular-frontend-frameworks-compared/> – Назва з екрана.
38. Pros and Cons of Bootstrap & Foundation – Know what you need! [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://webbmonks.com/blog/what-are-the-pros-cons-of-foundation-and-bootstrap/>– Назва з екрана.
39. Foundation. The most advanced responsive front-end framework in the world. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://foundation.zurb.com/>– Назва з екрана.
40. Semantic UI [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://semantic-ui.com/>– Назва з екрана.
41. Materialize [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://materializecss.com/>– Назва з екрана.
42. MATERIAL-UI [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://material-ui.com/>– Назва з екрана.
43. A lightweight and modular front-end framework for developing fast and powerful web interfaces. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: <https://getuikit.com/>– Назва з екрана.
44. В.М.Дубовой, С.М.Москвіна, О.Д.Никитенко. Моделювання процесів і систем керування з використанням UML
45. Usability Testing & Information Architecture [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2019 – Режим доступу: http://usabilitynet.org/trump/documents/Usability_standards.ppt.pdf– Назва з екрана.

46. 10 THINGS TO KNOW ABOUT COMPLETION RATES [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : MeasuringU 2011 – Режим доступу: <https://measuringu.com/blog/completion-rates/>– Назва з екрана.
47. Efficiency [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : UsabilityGeek 2019 – Режим доступу: <http://ui-designer.net/usability/efficiency.htm>– Назва з екрана.
48. Satisfaction [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Wikipedia 2019 – Режим доступу: <http://ui-designer.net/usability/satisfaction>– Назва з екрана.
49. Закон Ципфа [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : UsabilityGeek 2019 – Режим доступу: https://ru.wikipedia.org/wiki/Закон_Ципфа– Назва з екрана.
50. G. K. Zipf, *The Psycho-biology of Language: An Introduction to Dynamic Philology*. (Houghton Mifflin Company, 1935; MIT Press, 1965).
51. B. Mandelbrot, “An informational theory of the statistical structure of language”, in *Communication Theory*, ed. Willis Jackson. (Betterworths, 1953).
52. B. Mandelbrot, *The Fractal Geometry of Nature*. (Freeman, 1982).

ДОДАТКИ

Додаток А
(обов'язковий)
ВНТУ

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання магістерської кваліфікаційної роботи
Інформаційна система ієрархічної категоризації науковців за відкритими
даними з Google Scholar. Частина 2. Програмна реалізація системи та
експериментальні дослідження.

Студент групи 2АКІТ-18м Білоус Д.Д.

“ ___ ” _____ 2019 р.

Керівник д.т.н., професор Штовба С.Д.

“ ___ ” _____ 2019 р.

1. Назва та галузь застосування

1.1. Назва – «Інформаційна система ієрархічної категоризації науковців за відкритими даними з Google Scholar. Частина 2. Програмна реалізація системи та експериментальні дослідження.»

1.2. Галузь застосування – Інформаційні системи обробки та представлення інформації.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ № 238 від “28” “09” 2019 р.

3. Мета та призначення розробки.

Підвищення ефективності та якості процесу пошуку категоризованих науковців шляхом розробки інформаційної веб-системи ієрархічної категоризації науковців. Розробка інформаційної системи дозволить проводити пошук науковців за галузями науки, проводити їхнє рейтингування та автоматизувати трудомісткий процес класифікації.

4. Вихідні дані для проведення розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Handbook of Human-Computer Interaction – M.G. Helander, T.K. Landauer, P.V. Prabhu Elsevier, 18 серп. 1997 р. - 1582 стор.

2. В.М.Дубовой, С.М.Москвіна, О.Д.Никитенко. Моделювання процесів і систем керування з використанням UML

3. В. Mandelbrot, “An informational theory of the statistical structure of language”, in Communication Theory, ed. Willis Jackson. (Betterworths, 1953).

5.1. Перелік головних функцій:

- пошук науковця за прізвищем .
- пошук науковців за галузями науки.
- пошук науковців за спеціальностями.
- рейтингування науковців за показниками цитування.
- ієрархічна класифікація науковців

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- WINDOWS 7 та вище;
- JetBrains PhpStorm 2019.2.1 та вище.

5.2.2. Умови експлуатації системи:

- робота на настільних та портативних пристроях в приміщеннях зі стандартними умовами;
- можливість цілодобового функціонування системи;
- текст програмного забезпечення системи частково закритим.

6. Економічні показники

До економічних показників входять:

- коефіцієнта наукової значимості 0,52;
- розмір капіталовкладень не більше 52836 грн.;
- комерційний потенціал вище середнього;

7. Стадії та етапи розробки.

7.1 Пояснювальна записка:

- | | | |
|---|--|--------------|
| 1 | Огляд стану питання та постановка задач дослідження | 07.10.2019р. |
| 2 | Аналіз функцій системи та розробка архітектури системи | 15.10.2019р. |
| 3 | Аналіз структури системи та розробка структурної схеми системи | 22.10.2019р. |
| 4 | Проектування інтерфейсу системи | 29.10.2019р. |
| 5 | Розробка програмного забезпечення системи. | 05.11.2019р. |
| 6 | Тестування програмного забезпечення | 12.11.2019р. |
| 7 | Економічна частина | 19.11.2019р. |
| 8 | Оформлення пояснювальної записки, графічного матеріалу і презентації | 25.11.2019р. |
| 9 | Захист МКР | 12.12.2019р. |

7.2 Графічні матеріали:

- постановка задачі дослідження: «26»11.2019р.
- актуальність теми: «26»11.2019р
- UML-діаграма варіантів використання: «26»11.2019р
- UML-діаграма діяльності: «26»11.2019р
- UML-діаграма послідовності: «26»11.2019р
- інтерфейс системи: «26»11.2019р
- інтерфейс модального вікна системи: «26»11.2019р
- Тестування пошуку науковця: «26»11.2019р
- Тестування пошуку науковця. Модальне вікно: «26»11.2019р
- Тестування пошуку за галузями: «26»11.2019р
- Тестування рейтингування науковців: «26»11.2019р

8. Порядок контролю і приймання.

8.1. Хід виконання магістерської кваліфікаційної роботи контролюється керівником роботи, консультантами з економічної частини. Рубіжний контроль провести до «12» грудня 2019 р.

8.2. Атестація проекту здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «12» грудня 2019 р.

8.3. Підсумкове рішення щодо оцінки якості виконання магістерської кваліфікаційної роботи приймається на засіданні ДЕК. Захист магістерської кваліфікаційної роботи провести «12» грудня 2019 р.

Додаток Б

Лістинг програми

```

$(document).ready(function () {

    var baseUrl = "http://localhost/profilestatistic/public/";

    var currentScientistCount = 20;

    var maxLoad = 20;

    var scrolledDown = false;

    var subDispOptions = [];
    var scopusOptions = [];

    var filterRequest = {areas: [], disp: [], sort: []};
    var selectedFields = {areas: [], sub_disp: []};

    var sortOptions = {true: "asc", false: "desc"};

    var isSubsUnclassifiedSelected = false;
    var isAreasUnclassifiedSelected = false;
    var isUnclassified = false;
    var isAll = false;

    var firstOrderTypeSelectedVal = undefined;
    var secondOrderTypeSelectedVal = undefined;
    var thirdOrderTypeSelectedVal = undefined;
    var periodOfTime = undefined;
    var orderType = "DESC";

    var sortRequest = undefined;

    var scientistModalWindowInfo = document.getElementById('scientist_modal_window');
    var closeScientistInfo =
document.getElementsByClassName("close_scientist_info")[0];

    closeScientistInfo.onclick = onCloseScientistInfo;

    window.onclick = function (event) {
        if (event.target == scientistModalWindowInfo) {
            scientistModalWindowInfo.style.display = "none";
        }
    }

    var rangeItems = [];
    rangeItems[0] = "Low";
    rangeItems[1] = "Lower Average";
    rangeItems[2] = "Average";
    rangeItems[3] = "Higher Average";
    rangeItems[4] = "High";

    var initialValues = [0],
    updateValue = function (ui, id, fieldType) {
        var rangeItem = Math.floor(ui.value);
        console.log(id);
        $(ui.handle).attr('data-value', rangeItems[rangeItem]);
        $(ui.handle).attr('data-number', id);
        var idx = _.findIndex(selectedFields[fieldType], {'id': String(id)});

```

```

        console.log(idx);
        if (idx > -1) {
            selectedFields[fieldType][idx].range = rangeItems[rangeItem];
        }
    };

function onCloseScientistInfo() {
    scientistModalWindowInfo.style.display = "none";
}

// Find object in array: let index = _.findIndex(list, { 'num': `145K` });
function createFuzzyRange(id, idNumber, fieldType) {
    // From: http://jsfiddle.net/Mottie/vc8tumd9/5/
    $('#${id}`).slider({
        min: 0,
        max: 4,
        step: 1,
        range: false,
        values: [0],
        create: function (event, ui) {
            $.each(initialValues, function (i, v) {
                updateValue({
                    value: v,
                    handle: $(''.ui-slider-handle').eq(i)
                }, idNumber);
            });
        },
        slide: function (event, ui) {
            updateValue(ui, idNumber, fieldType);
        }
    });
}

var scientistAutoComplete = new autoComplete({
    selector: 'input[id="scientist_auto_complete"]',
    minChars: 2,
    source: function (term, suggest) {
        term = term.toLowerCase();
        $.ajax({
            url: baseUrl + "autocomplete_scientist_name",
            type: "post",
            data: {"scientist_name": term}
        })
        .done(function (data) {
            let names = data.map(value => value.name);
            suggest(names);
        });
    }
});

$("#scientist_auto_complete").on("keydown", function (eventObj) {
    if (eventObj.keyCode === 13) {
        sendReqToObtainScientist($("#scientist_auto_complete").val());
    }
});

console.log("test");
loadAreas();

console.log("test");
loadScientistFromFilter(maxLoad, 0, selectedFields, function () {
});

```

```

$("#sort_scientist_action").prop("disabled", true);

$("#sort_scientist_action").on("click", function () {
    $("#filtered_scientists").html("");
    $("#sort_scientist_action").prop("disabled", true);
    currentScientistCount = maxLoad;
    loadScientistFromFilter(maxLoad, 0, selectedFields, function () {
        $("#sort_scientist_action").prop("disabled", false);
        $("#sort_scientist_action").css("cursor", "pointer");
    });
});

$("#order_type").change(function () {
    orderType = $(this).val();
});

$("#first_order_type").change(function () {
    let value = $(this).val();

    firstOrderTypeSelectedVal = value;

    $('#second_order_type option').prop("disabled", false);
    $('#second_order_type option:contains(${value})').prop("disabled", true);
    $('#second_order_type').prop('selectedIndex', 0);
    $("#second_order_type").prop('disabled', false);

    $('#third_order_type option:contains(${value})').prop("disabled", true);
    $('#third_order_type').prop('selectedIndex', 0);

    $("#third_order_type").prop("disabled", true);
    $("#sort_scientist_action").prop("disabled", true);
});

$("#second_order_type").change(function () {
    var value = $(this).val();
    secondOrderTypeSelectedVal = value;

    $('#third_order_type option').prop("disabled", false);
    $('#third_order_type
option:contains(${firstOrderTypeSelectedVal})').prop("disabled", true);
    $('#third_order_type
option:contains(${secondOrderTypeSelectedVal})').prop("disabled", true);
    $('#third_order_type').prop('selectedIndex', $('#third_order_type
option:enabled')[1].index);

    thirdOrderTypeSelectedVal = $('#third_order_type option:enabled')[1].text;

    $("#third_order_type").prop('disabled', false);
    $("#sort_scientist_action").prop("disabled", false);
});

$("#period_of_time").change(function () {
    var value = $(this).val();
    periodOfTime = value;
});

$(document).keyup(function (e) {
    if (e.keyCode == 27) {
        onCloseScientistInfo();
    }
});

```

```

$("#find_scientist").on("click", function () {
    sendReqToObtainScientist($("#scientist_auto_complete").val());
});

$("#filter_scientist_action").on("click", function () {
    $(this).prop("disabled", true);
    $(this).css("cursor", "no-drop");
    $("#filtered_scientists").html("");
    currentScientistCount = maxLoad;
    loadScientistFromFilter(maxLoad, 0, selectedFields, function () {
        $("#filter_scientist_action").prop("disabled", false);
        $("#filter_scientist_action").css("cursor", "pointer");
    });
});

/**
 * Галузі науки
 */
function loadAreas() {
    $.ajax({
        url: baseUrl + "DimensionsFields",
        type: "post",
        success: function (data) {
            representFields(data, "scopus_list", selectedFields, "areas");
        },
        error: function (err) {
            console.log(err);
        }
    });
}

// todo review
function representSpringer(id, fields) {
    for (let i = 0; i < fields.length; i++) {
        $('#${id} .areas_list').append(makeSpringerItem(fields[i].id,
fields[i].name));
        createFuzzyRange(`range-sp-${fields[i].id}`, fields[i].id, "sub_disp");
        $('#' + id + " .areas_list .ui-slider").hide();
    }

    $('#${id} .areas_list a').on("click", function (event) {
        var $target = $(event.currentTarget),
            val = $target.attr('data-value'),
            $inp = $target.find('input'),
            fieldType = "sub_disp",
            idx;

        if ((idx = _.findIndex(selectedFields[fieldType], {'id': val})) > -1) {
            selectedFields[fieldType].splice(idx, 1);

            $(this).siblings(".ui-slider").hide();
            setTimeout(function () {
                $inp.prop('checked', false)
            }, 0);
        } else {
            $(this).siblings(".ui-slider").show();

            selectedFields[fieldType].push({"id": val, "range": "Low"});
            setTimeout(function () {
                $inp.prop('checked', true)
            }, 0);
        }
    }
}

```

```

        console.log(JSON.stringify(selectedFields));
        $(event.target).blur();
        return false;
    });
}

function representFields(fields, id, selected, fieldType) {

    for (let i = 0; i < fields.length; i++) {
        $("#" + id + " .areas_list").append(makeScopusItem(fields[i].id,
fields[i].name, fields[i].subFields));
        createFuzzyRange(`range-${fields[i].id}`, fields[i].id, "areas");
        $('#customControlInline${fields[i].id}`).on('change', function () {
            $(this).siblings(".subareas").toggle();
            console.log("on change")
        });
        $("#" + id + " .areas_list .ui-slider").hide();
        $("#" + id + " .areas_list .subareas").hide();
    }
    var fieldType = "areas";
    // todo review
    $('#${id} .areas_list a').on("click", function (event) {
        var $target = $(event.currentTarget),
            val = $target.attr('data-value'),
            $inp = $target.find('input'),
            idx;
        let checked = $(this).prop('checked');
        if (checked === false) {
            $(this).siblings(".subareas").show();
            $(this).prop('checked', true);
        } else {
            $(this).siblings(".subareas").hide();
            $(this).prop('checked', true);
        }
        console.log("on click" + checked);
        if (checked) {
        } else {
        }
    });

    // $(event.target).blur();
    return false;
});

$('#${id} .areas_list').on('focusout', function () {
    findSubDisciplineByScopusIds(selectedFields[fieldType]);
});
}

function makeScopusItem(id, name, fields) {
    return `
        <div class="check custom-control custom-checkbox" tabIndex="-1">
            <input type="checkbox" class="custom-control-input"
id="customControlInline${id}" data-value="${id}">
            <label class="custom-control-label "
for="customControlInline${id}" data-value="${id}">${name}</label>
            <div id="range-${id}"></div>
            <div class="check custom-control custom-checkbox
subareas">${fields.map(f => `${makeSubFieldItem(f)}`).join("")}</div>
            </div>`;
}

function makeSubFieldItem(subField) {

```

```

        return `
```



```

        "subFields": filter.sub_disp,
        "firstOrder": firstOrderTypeSelectedVal,
        "secondOrder": secondOrderTypeSelectedVal,
        "thirdOrder": thirdOrderTypeSelectedVal,
        "periodOfTime": periodOfTime,
        "sortValue": orderType,
        "isUnclassified": isUnclassified,
        "isAll": isAll
    })),
    success: function (data) {
        $('#loading_progress').hide();
        callBack();
        console.log(data);

        const scientists = data.data;
        if (scientists.length > 0) {
            traverseOverScientists(scientists, "#filtered_scientists");
        } else {
            $("#filtered_scientists").append("<h4 style='margin:auto'>Not
found</h4>");
        }
    },
    error: function (err) {
        $('#loading_progress').hide();
        callBack();
        console.log(err);
    }
});
}

function sendReqToObtainScientist(scientistName) {
    $.ajax({
        url: baseUrl + "find_scientist_by_name",
        type: "post",
        data: {"scientist_name": scientistName},
        success: function (data) {
            var scientist = jQuery.parseJSON(data);

            if (scientist !== null) {
                representFoundScientist(scientist, "found_scientist_info");
            }
        },
        error: function (err) {
            console.log(err);
        }
    });
}

function representFoundScientist(scientist, id) {
    let springerStats = jQuery.parseJSON(scientist.dimensionsSubField);
    let scopusStats = jQuery.parseJSON(scientist.dimensionsField);

    $('##{id}`).show();

    representScientistInfo(id, scientist, springerStats, scopusStats);
}

function representScientist(scientist, containerId) {
    let springerStats = scientist.dimensionsSubField;
    let scopusStats = scientist.dimensionsField;

    representShortedScientistInfo(containerId, scientist);
}

```

```

    $('#${containerId}`).on("click", function () {
        showScientistInfoInModal(scientist, springerStats, scopusStats);
    });
}

function showScientistInfoInModal(scientist, springerStats, scopusStats) {
    scientistModalWindowInfo.style.display = "block";
    representScientistInfo("scientist-info-content", scientist, springerStats,
scopusStats);
}

function representShortedScientistInfo(id, scientist) {
    $('#${id}`).html("");

    $('#${id}`).html(templateForShorterScientiest(scientist));
}

/**
    Інформація науковця
    */
function templateForShorterScientiest(scientist) {
    return `
        <div class="scientist_info">
            <div class="common_info row">
                <div class="scientist_photo col-auto col-sm-
auto col-md-auto col-lg-auto">
                    
                </div>

                <div class="text_info col">
                    <p
class="scientist_name">${scientist.name}</p>

                    <p class="scientist_organization">Інститут
сцинтиляційних матеріалів НАН України, Institute for Scintillation Materials</p>
                </div>

                <div class="citations_box d-flex col- col-sm-
auto col-md-auto col-lg-auto justify-content-end">
                    <div class="scientist_citations">
                        
                        <div class="w-100"></div>
                        <span
class="citations_value">${scientist.citations[0].value}</span>
                    </div>
                    <div class="scientist_h_index">
                        
                        <div class="w-100"></div>
                        <span
class="citations_value">${scientist.h_index[0].value}</span>
                    </div>
                    <div class="scientist_i10_index">
                        
                        <div class="w-100"></div>
                        <span
class="citations_value">${scientist.i10_index[0].value}</span>
                    </div>
                </div>
            </div>
        `;
}

```

```

        </div>

        </div>
        <div class="scientist_interests">
    ${templateForScientistInterestsItem(scientist.specialities)}
        </div>
    </div>

    `;
}

function formatShortedScholarOptions(scholarOption, name) {
    return `<table class="table">
        <th>${name}</th>
        ${scholarOption ? mapToCorrect(scholarOption).map(scholarValue =>
`<td>${scholarValue.value}</td>`)}.join("") : ``}
        </table>`;
}

function templateForScientistInterestsItem(interests) {
    return `
        ${interests.map(interest => `<div
class="scientist_interest_item">${interest.name}</div>`)}.join("")}
        `;
}

function templateForScientistInterests(interests) {
    return `
        ${interests.map(interest => `<div
class="scientific_interests_item">${interest.name}</div>`)}.join("")}
        `;
}

function formatScholarOptions(scholarOption, name) {
    return `<tr class="table_row">
        <th scope="row" class="table_first_col">${name}</th>
        ${scholarOption ? mapToCorrect(scholarOption).map(scholarValue => `<td
class="table_col">${scholarValue.value}</td>`)}.join("") : ``}
        </tr>`;
}

function templateForCategoriesLegend(labels, colors, values) {
    return `<ul class="items_legend">
        ${labels.map((label, labelId) => `<li style="color:
${colors[labelId]}">${values[labelId]}% - ${label}</li>`)}.join("")}
        </ul>`;
}

/**
    Модальне вікно
    */
function scientistInfoTemplate(scientist) {
    return `<div id="scientist-info-content" class="main-content">
        <div class="modal-header d-flex justify-content-start">
            
            <div class="modal_header_text ">
                <p class="modal_scientist_name">${scientist.name}</p>
                <a class="scholar_link" href="${scientist.Scholar}" >Profile
in Google Scholar</a>

```

```

        <p class="modal_scientist_organization">Інститут
сцинтиляційних матеріалів НАН України, Institute for Scintillation Materials</p>
    </div>
</div>
<div class="modal-body row">
<div class="col-lg-12 ">
    <div class="scientific_interests_box">
        <div id="scientific_interests_header" class="modal_box_header d-
inline-flex justify-content-start">
            
            <div class="modal_box_title ">Scientific interests</div>
        </div>
        <div class="scientific_interests_list d-inline-flex
justify-content-start">
            ${templateForScientistInterests(scientist.specialities)}
        </div>
    </div>
</div>
</div>
    <div class="col-lg">
    <div class="scholar_indicators_box ">
<div class="modal_box_header d-inline-flex justify-content-
start">
    
    <div class="modal_box_title ">Google Scholar
indicators</div>
    </div>
    <table id="indexes_table" class="table">
    <tbody class="table_body">
    <tr class="table_row">
    <th scope="col" class="table_col table_top_row"></th>
    <th scope="col" class="table_col
table_top_row">All</th>
    <th scope="col" class="table_col table_top_row">Since
2013</th>
    </tr>
    </tbody>
    <tbody class="table_body">
    <tr class="table_row">
    <td colspan="3">
        ${formatScholarOptions(scientist.citations,
"quotations")}
        <td colspan="3">
            ${formatScholarOptions(scientist.h_index, "h-index")}
            ${formatScholarOptions(scientist.i10_index, "i10-
index")}
        </td>
    </tr>
    </tbody>
    </table>
    </div>
</div>
<div class="col-lg">
    <div class="areas_chart_box">
<div class="modal_box_header d-inline-flex justify-content-
start">
    
    <div class="modal_box_title ">Areas</div>
    </div>
    <div class="scopus_items">
    <div class="">
    <canvas id="areas_${scientist.id}" class="chartjs-render-
monitor" style="display: block;"></canvas>

```

```

        </div>

        <div class="legend-items"
id="scopus_legend_${scientist.id}"></div>
    </div>
    </div>
    </div>
    <div class="col-lg">
    <div class="subareas_chart_box">
    <div class="modal_box_header d-inline-flex justify-content-
start">
        
        <div class="modal_box_title ">Subareas</div>
        </div>
        <div class="springer_items">
        <div class="">
            <canvas id="sub_disp_${scientist.id}" width="300"
height="150" class="chartjs-render-monitor" style="display: block;"></canvas>
        </div>
        <div class="w-100"></div>
        <div class="legend-items"
id="springer_legend_${scientist.id}"></div>
        </div>
        </div>
        </div>
    </div>
    </div>`;
}

function representScientistInfo(id, scientist, springerStats, scopusStats) {
    $(`#${id}`).html("");

    $(`#${id}`).html(scientistInfoTemplate(scientist));

    if ((scopusStats === undefined
        && springerStats === undefined) ||
        (springerStats.length === 0 && scopusStats.length === 0)) {
        $(`#categorization_message_${scientist.id}`).hide();
    } else {
        $(`#categorization_message_${scientist.id}`).show();
    }

    representStats("areas_" + scientist.id, scopusStats, "",
`scopus_legend_${scientist.id}`);
    representStats("sub_disp_" + scientist.id, springerStats, "",
`springer_legend_${scientist.id}`);
}

function mapToCorrect(array) {
    return array.map(function (val) {
        if (!("value" in val)) {
            return {"value": 0};
        }
        return val;
    });
}

function representStats(id, stats, name, legendId) {
    let labels = [];
    let values = [];
    let colors =

```

```
["#91a0d9", "#7487cf", "#576ec5", "#3f58b6", "#354a99", "#2b3c7b", "#212d5e", "#161f41", "#0c1124", "#020307"];
```

```

    for (let i = 0; i < stats.length; i++) {
      labels.push(stats[i].name);
      values.push((stats[i].value).toFixed(2));
      colors.push(getRandomColor());
    }

    $('#${legendId}`).html('');
    $('#${legendId}`).html(templateForCategoriesLegend(labels, colors, values));

    drawBar(id, labels, values, colors, name);
  }

function drawBar(containerId, labels, values, colors, title) {
  let parent = $('#${containerId}`).parent();
  $('#${containerId>').remove();
  parent.append('<canvas id="' + containerId + '"></canvas>');

  let canvasElem = document.getElementById(containerId);
  if (canvasElem === null) {
    return;
  }

  let ctx = canvasElem.getContext('2d');
  let maxValue = values.reduce(function (a, b) {
    return Math.max(a, b);
  });
  let chart = new Chart(ctx, {
    type: 'doughnut',
    data: {
      labels: labels,
      datasets: [{
        data: values,
        backgroundColor: colors
      }]
    },
    options: {
      responsive: true,
      scaleBeginAtZero: true,
      legend: {
        position: 'top',
        display: false
      },
      title: {
        display: false,
        text: title
      },
      scales: {
        yAxes: [{
          display: false
        }],
        xAxes: [{
          display: false
        }]
      }
    }
  });
}

function traverseOverScientists(scientists, containerId) {
  for (let i = 0; i < scientists.length; i++) {

```

```
        let scientistId = "scientist_" + scientists[i].id;
        $(containerId).append(templateForScientistContainer(scientistId));
        representScientist(scientists[i], scientistId)
    }
}

function templateForScientistContainer(scientistId) {
    return `
```

Додаток В
(Обов'язковий)

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
д.т.н., проф. В.М. Дубовой

« _____ » _____ 2019 р.

ПЕРЕЛІК
ГРАФІЧНИХ МАТЕРІАЛІВ

для захисту магістерської кваліфікаційної роботи
на тему

ІНФОРМАЦІЙНА СИСТЕМА ІЄРАРХІЧНОЇ КАТЕГОРИЗАЦІЇ
НАУКОВЦІВ ЗА ВІДКРИТИМИ ДАНИМИ З GOOGLE SCHOLAR. ЧАСТИНА
2. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ТА ЕКСПЕРИМЕНТАЛЬНІ
ДОСЛІДЖЕННЯ.

1. Постановка задачі дослідження.
2. Актуальність теми.
3. UML-діаграма варіантів використання.
4. UML-діаграма діяльності.
5. UML-діаграма послідовності.
6. Інтерфейс системи
7. Інтерфейс модального вікна веб-системи
8. Тестування пошуку науковця
9. Тестування пошуку науковця. Модальне вікно
10. Тестування пошуку за галузями
11. Тестування рейтингування науковців

Розробив: Білоус Д.Д.

_____ (підпис) _____ (дата)

Перевірив: Штовба С.Д.

_____ (підпис) _____ (дата)

Рецензент: Софіна О.Ю.

_____ (підпис) _____ (дата)

1. Постановка задачі дослідження.

Мета роботи: : підвищення ефективності та якості процесу пошуку категоризованих науковців шляхом розробки інформаційної веб-системи ієрархічної категоризації науковців.

Об'єкт дослідження: інформаційні системи підтримки наукової діяльності

Предмет дослідження: інформаційна технологія ієрархічної категоризації науковців на основі їх інтересів.

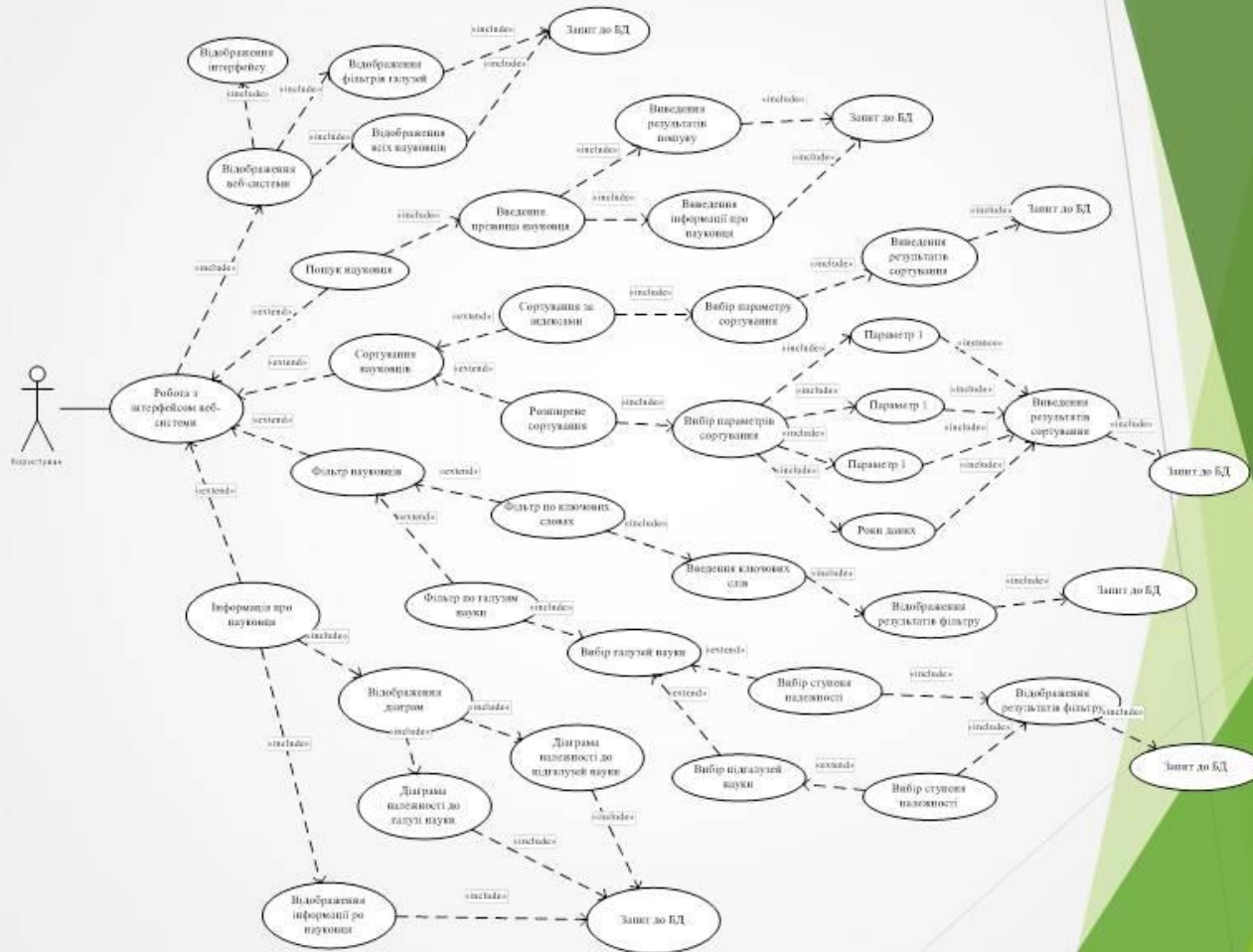
2. Актуальність теми.

Актуальність теми

- ▶ В сучасному світі люди повсякчас стикаються із задачею пошуку інформації. Наразі більша частина людства постійно користується мережею Інтернет для пошуку тієї чи іншої інформації. Можливість пошуку в мережі Інтернет забезпечують інформаційно-пошукові системи найбільш очевидними прикладами яких можна назвати зокрема такі сервіси як Yahoo та Google. Більшість таких систем відповідають за пошук широкого набору інформації а також існують системи які дозволяють здійснювати пошук більш спеціалізованої інформації. Однак на даний момент немає систем які дозволяли б здійснювати пошук науковців які є класифікованими за певною галуззю чи підгалуззями і при цьому надавали широкий набір інформації про кожного науковця.
- ▶ Розробка інформаційної системи дозволить проводити пошук науковців за галузями науки, проводити їхнє рейтингування та автоматизувати трудомісткий процес класифікації.

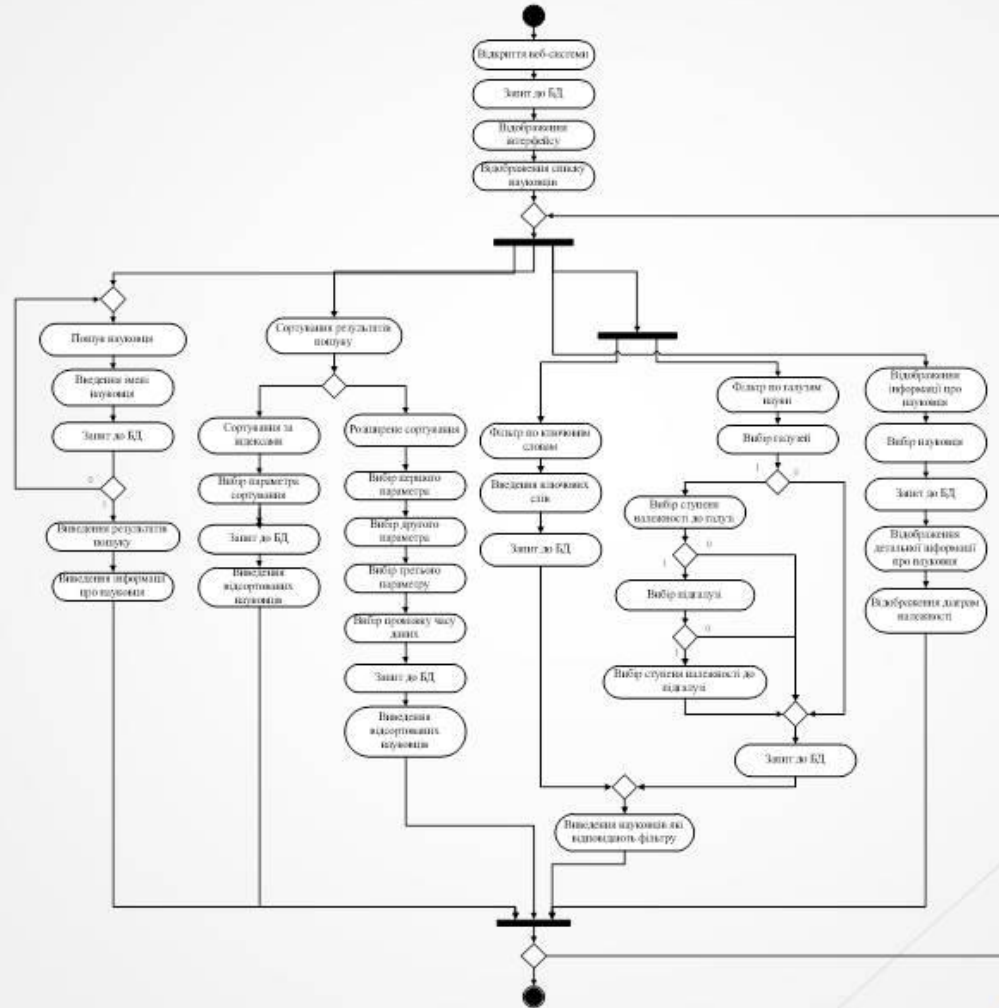
4. UML-діаграма варіантів використання.

Діаграма use-case



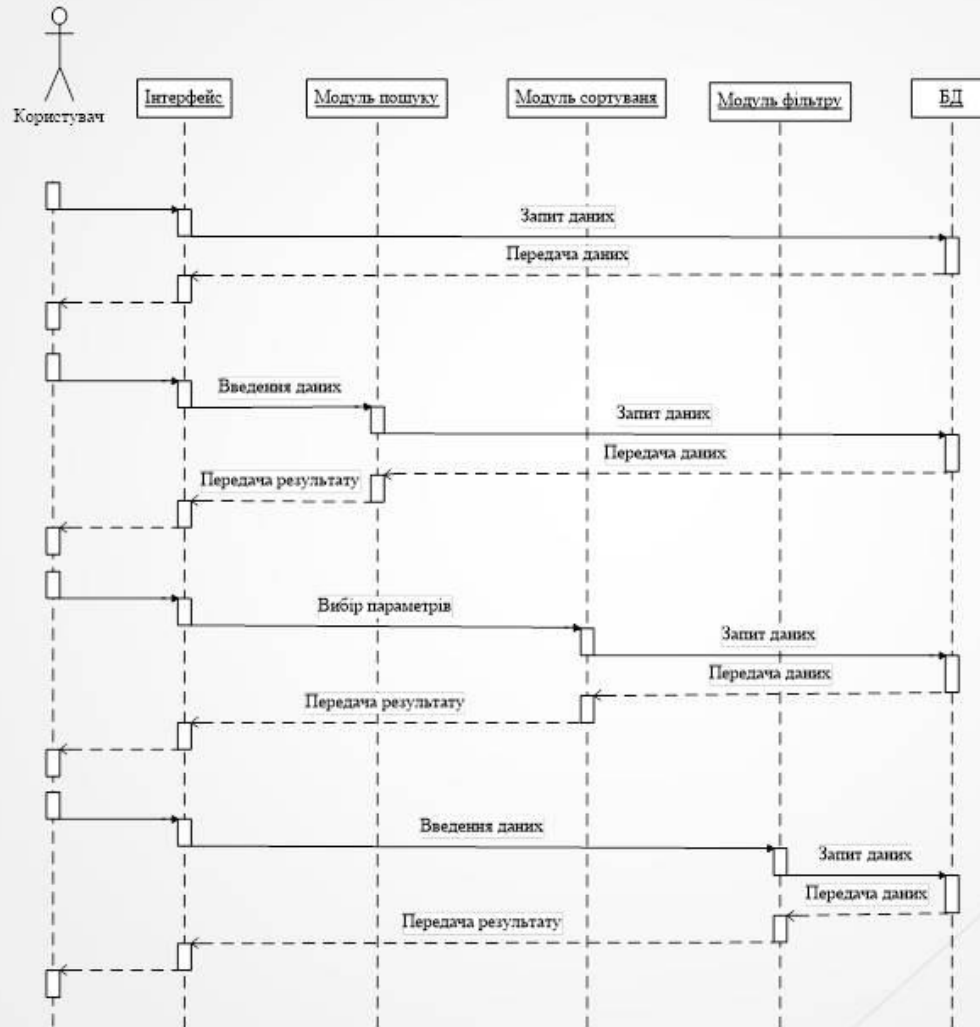
3. UML-діаграма діяльності.

UML-діаграма діяльності



4. UML-діаграма послідовності.

UML-діаграма послідовності



5. Інтерфейс системи

Інтерфейс веб-системи

ASC AUTOMATED SCIENTIST CLASSIFICATOR HOW IT WORKS ABOUT US RESPONSE

Search

FILTERS Clear all

Key words Clear

medicine × physics ×
instrumentation ×

Areas Clear

Agricultural and Veterinary Sciences

Biological Sciences

Biochemistry and Cell Biology

Ecology

Evolutionary Biology

Genetics

Built Environment and Design

Chemical Sciences

Commerce, Management, Tourism and Services

Earth Sciences

Economics

Sort by Citations Advanced sorting

Борис Гриньов 51358 89 298

Державний фонд фундаментальних досліджень,
Інститут сцинтиляційних матеріалів НАН України

solid state physics materials instrumentation
science of science

Evgenij Martynov 17491 66 114

Bogolyubov Institute for Theoretical Physics,
National Academy of Sciences, Ukraine

physics


Igor Bondarenko 16535 43 95

Professor of Oncology, Dnipropetrovsk Medical
Academy (Ukraine)

medicine clinical oncology cancer research clinical trials

5. Інтерфейс модального вікна веб-системи

Інтерфейс модального вікна веб-системи



Борис Гриньов
Profile in Google Scholar

Державний фонд фундаментальних досліджень, Інститут скінтіляційних матеріалів НАН України

✕


■ SCIENTIFIC INTERESTS

solid state physics materials instrumentation science of science

■ GOOGLE SCHOLAR INDICATORS


	All	Since 2013
citations	51358	44249
h-index	89	87
i10-index	298	267

■ AREAS



79.16% Other Physical Sciences
20.84% Materials Engineering

■ SUBAREAS



79.16% Other Physical Sciences
20.84% Materials Engineering

6. Тестування пошуку науковця

Тестування пошуку науковця

Scientist name

чека

- Валентина Чеканова
- Галина Чекаловська
- юрій чекан
- Николай Анатольевич Чеканов, Микола Анатолійович Чеканов, M. Chekanov
- Ольга Чекамова

[Advanced sorting](#)

Sort by

- Chekalenko Lyudmyla D (Чекаленко Людмила Дмитрівна)

9. Тестування пошуку науковця. Модальне вікно

Тестування пошуку науковця. Модальне вікно

Інститут сцинтиляційних матеріалів НАН України, Institute for Scintillation

Chekalenko Lyudmyla D (Чекаленко Людмила Дмитрівна)
 Profile in Google Scholar
 Taras Shevchenko National University of Kyiv; Дипломатична академія України при МЗС України

SCIENTIFIC INTERESTS | foreign policy of ukraine international relations / foreign policy international narrative

	All	Since 2013
citations	261	156
h-index	7	5
i10-index	5	2

AREAS

100.00% - Studies in Human Society

SUBAREAS

100.00% - Political Science

10. Тестування пошуку за галузями

Тестування пошуку за галузями

ASC AUTOMATED SCIENTIST CLASSIFICATOR ABOUT US HOW IT WORKS RESPONSE

Scientist name Search

Sort by h-index [Advanced sorting](#)

FILTERS Clear all

Key words Clear

Key word

Areas Clear

Agricultural and Veterinary Sciences

Agriculture, Land and Farm Management

Animal Production

Crop and Pasture Production













Fisheries Sciences

Forestry Sciences

Horticultural Production

Veterinary Sciences

Other Agricultural and Veterinary Sciences

	Юлія Пославська Julija Poslavska Stepan Gzhytskyi National University of Veterinary Medicine and Biotechnologies Lviv genetics	 28	 3	 0
	Постернак Леонід Іванович / Posternak Leonid / Постернак Л. І. Вінницький національний аграрний університет, кандидат с.-г. наук, доцент	 5	 2	 0
	Налія Анатоліївна Меженська старший науковий співробітник ДНДІЛДВСЕ	 6	 2	 0

12. Тестування рейтингування науковців

Тестування рейтингування науковців

ASC AUTOMATED SCIENTIST CLASSIFICATOR ABOUT US HOW IT WORKS RESPONSE

Scientist name

Sort by [Advanced sorting](#)


FILTERS

Key words

Key word


Areas

- Agricultural and Veterinary Sciences
- Biological Sciences
- Built Environment and Design
- Chemical Sciences
- Commerce, Management, Tourism and Services
- Earth Sciences
- Economics
- Education
- Engineering
- Environmental Sciences
- History and Archaeology
- Information and Computing

 **Vladimir Korobov (Володимир Коробов)**

Херсонський державний університет, факультет біології, географії і екології, кафедра соціально

415 **4** **3**

 **Маслюченко Володимир Кирилович / Volodymyr K. Maslyuchenko**

Завідувач кафедри математичного аналізу ЧНУ імені Юрія Федьковича

392 **8** **7**