

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

**Реалізація інтерфейсу користувача із застосуванням багатовимірного
адресного сортування**

**Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки» - 122**

Керівник курсової роботи

доцент

Ющенко Ю. О.

(Підпис)

“ ____ ” _____ 2022 року

Виконав студент КН-3

Кириченко С. В.

“ ____ ” _____ 2022 року

Київ – 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем

_____ Жежерун О.П.

“__” _____ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу студентці Кириченко Світлані Віталіївні
факультету інформатики 3 курсу бакалаврської програми

**ТЕМА: Реалізація інтерфейсу користувача із застосуванням
багатовимірного адресного сортування**

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

Теоретична частина

Проектування додатку

Висновки

Дата видачі „__” _____ 2022 р. Керівник _____

(підпис)

Завдання отримав _____ (підпис)

Студент Кириченко С. В.

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	13.10.2021	
2.	Аналіз актуальності, визначення недоліків і переваг	15.11.2021.	
3.	Вивчення технологій для розробки системи.	25.11.2021.	
4.	Створення технічного завдання.	28.06.2022.	
5.	Написання практичної частини.	09.07.2022.	
6.	Написання теоретичної частини.	13.07.2022.	
7.	Внесення змін до роботи відповідно до зауважень наукового керівника.	16.07.2022.	
9.	Захист курсової роботи.	24.07.2022	

Керівник Ющенко Ю. О.

“ ____ ” _____ 2022 р.

Зміст

Анотація	5
Вступ	6
РОЗДІЛ 1: Багатовимірний і одновимірний користувацький інтерфейс	8
1.1 Поняття багатовимірного і одновимірного інтерфейсу	8
1.2 Переваги і недоліки багатовимірного користувацького інтерфейсу	9
1.2.1 Переваги багатовимірного користувацького інтерфейсу	9
1.2.1 Недоліки багатовимірного інтерфейсу пошуку	11
1.3 Порівняльна характеристика одновимірного і багатовимірного інтерфейсу	12
1.3 Висновки до розділу 1.....	13
РОЗДІЛ 2: Багатовимірного адресне сортування	15
2.1 Поняття багатовимірного адресного сортування	15
2.2 Підходи до збереження даних для багатовимірного адресного сортування	15
2.2.1 Збереження даних за допомогою масивів індексації	16
2.2.2 Збереження даних за допомогою двозв'язних списків	17
2.2.3 Збереження даних за допомогою зовнішніх ключів	19
2.2.4 Збереження даних за допомогою кластеризації бази даних	20
2.3 Порівняльна характеристика підходів до варіанти збереження даних, для багатовимірного адресного сортування	21
2.4 Висновки до розділу 2.....	22
РОЗДІЛ 3: Проектування і програмна реалізація	24
3.1 Обґрунтування вибраної предметної області	24
3.2 Складові багатовимірного користувацького інтерфейсу	24
3.3 Стек технологій	27
3.4 Бази даних	28
3.4.1 База даних для багатовимірного адресного сортування	28
3.4.1 База даних для збереження фотографій	29
3.5 Взаємозв'язок застосунку з базами даних	30
3.6 Шляхи покращення застосунку	31
3.7 Висновки до розділу 3.....	32
Висновки	33
Список використаної літератури.....	34

Анотація

У курсовій роботі були розглянуті переваг і недоліків багатовимірного користувацького інтерфейсу, було порівняно підходи до реалізації багатовимірного адресного сортування.

У роботі запропоновано і наведено приклад імплементації методу реалізації багатовимірного адресного сортування за допомогою кластеризованої бази даних.

Ключові слова: багатовимірний інтерфейс, користувацький інтерфейс, адресне сортування, списки, двозв'язні списки, індексація, кластеризація, бази даних.

Вступ

Сучасний світ надзвичайно конкурентний і аби закріпитися на позиції потрібно невідпинно рухатися і покращувати якісь послуг. Це стосується і сфери пошукових програм. Одна із складових пошукової системи - це користувацький інтерфейс.

Відповідно до визначення, користувацький інтерфейс(UI) – це “це місце взаємодії та комунікації людини з комп’ютером”.[1] Користувацький інтерфейс створює форму для інформації, яку потрібно надати користувачу. Він нього залежить популярність системи, оскільки користувачі хочуть бачити інтуїтивно зрозумілий, зручний і привабливий інтерфейс, хочуть приділяти час на обробку інформації, яку надає застосунок, а не на розбирання з тим, як цю її отримати або прочитати.

Одним з видів користувацького інтерфейсу є багатовимірний інтерфейс.

Актуальність та практичне значення обраної теми

Багатовимірний інтерфейс надає можливість відображати багатовимірні дані. Користувач багатовимірного інтерфейсу який надає користувачу має можливість гортати список елементів у n напрямках, що відкриває купу можливостей.

На практиці багатовимірний користувацький інтерфейс може покращити застосунок у сфері продажу, або стати інтерактивним підручником, наприклад, з біології, в якому можна було б переходити від одного виду до іншого.

На жаль, не зважаючи на усі свої переваги, такий вид користувацького інтерфейсу поки що недостатньо популярний і досліджений.

Актуальність цієї курсової роботи зумовлена дослідженням переваг і недоліків багатовимірного користувацького інтерфейсу у порівнянні зі звичним одновимірним а також виявленням нового підходу до реалізації багатовимірного адресного сортування і порівнянням підходів до реалізації

багатовимірного адресного сортування, яке лежить в основі створення багатовимірного користувацького інтерфейсу.

Мета та завдання курсової роботи

Мета: Проаналізувати переваги і недоліки багатовимірного інтерфейсу і способи його імплементації.

Завдання: Дослідження переваги і недоліки багатовимірного інтерфейсу, створення власної реалізації багатовимірного інтерфейсу.

Об'єкт дослідження

Об'єктом дослідження курсової роботи є багатовимірний інтерфейс

Методи дослідження

Дослідження переваги і недоліки багатовимірного інтерфейсу буде здійснюватися за допомогою його порівняльного аналізу з одновимірним інтерфейсом.

Визначення переваг і недоліків реалізації багатовимірного адресного сортування буде здійснюватися за допомогою його порівняльного аналізу різних підходів до нього.

Джерела дослідження

Було вивчено дослідження, які існують на тему багатовимірного інтерфейсу.

РОЗДІЛ 1: Багатовимірний і одновимірний користувацький інтерфейс

У цьому розділі означимо поняття багатовимірного і одновимірного інтерфейсу і шляхом проведення порівняльної характеристики одновимірного інтерфейсу з багатовимірним визначимо його недоліки і переваги.

1.1 Поняття багатовимірного і одновимірного інтерфейсу

Одновимірний користувацький інтерфейс – це користувацький інтерфейс, який надає користувачу можливість рухатися вздовж однієї осі характеристики, збільшуючи чи зменшуючи її значення. Може включати в себе можливість додавати фільтри і змінювати характеристику, за якою здійснюється рух.

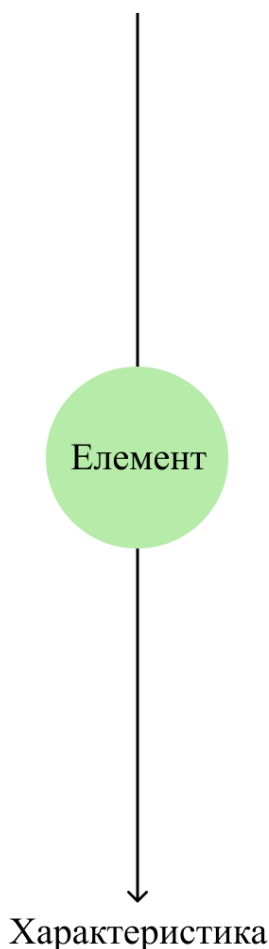


рис. 1. Схематичне зображення одновимірного користувацького інтерфейсу

Натомість, багатовимірний користувацький інтерфейс – це користувацький інтерфейс, який надає користувачу можливість рухатися по n осям характеристик, відповідно збільшуючи або зменшуючи одну з них,

лишаючи інші, за можливості, незмінними. Додаткові виміри розміщуються симетрично і рівновіддалено від сусідніх. Може включати в себе можливість додавати фільтри і змінювати характеристики, за якими здійснюється рух.

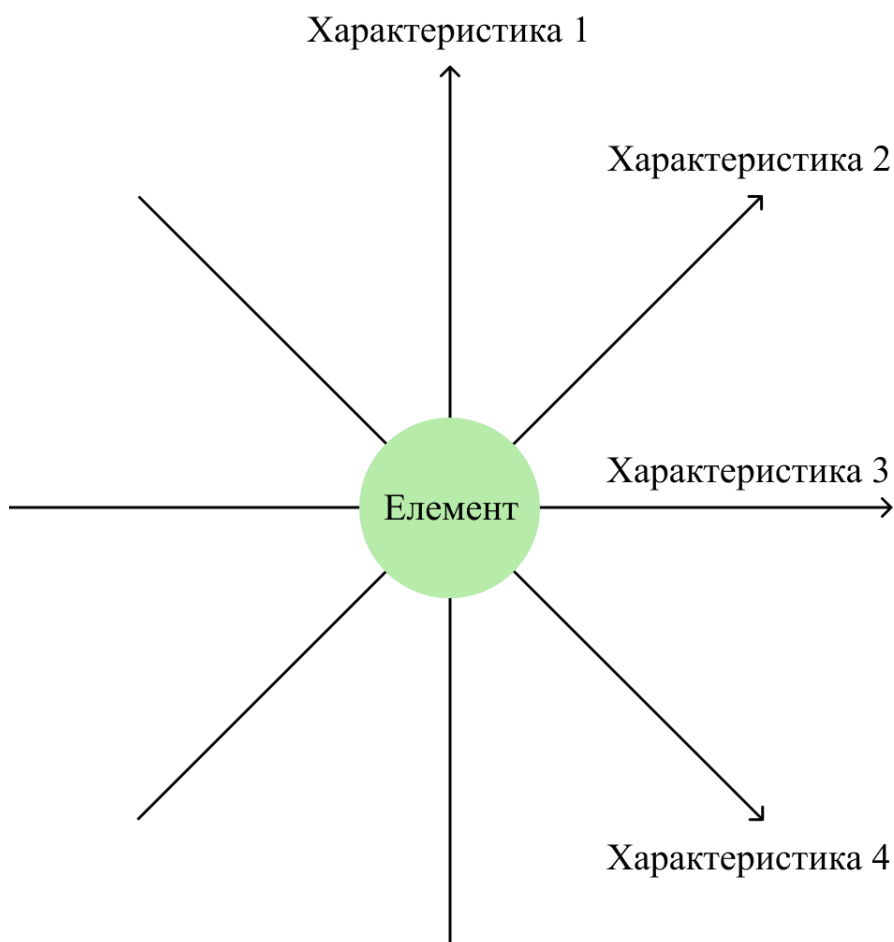


рис. 2. Схематичне зображення багатовимірного користувацького інтерфейсу

Осьова характеристика – це характеристика, значення якої змінюється вздовж однієї з осі.

В наступному параграфі описуються переваги і недоліки багатовимірний інтерфейсу перед одновимірним.

1.2 Переваги і недоліки багатовимірного користувацького інтерфейсу

Спочатку перелічимо переваги багатовимірного користувацького інтерфейсу над одновимірним.

1.2.1 Переваги багатовимірного користувацького інтерфейсу

Основна перевага багатовимірного користувацького інтерфейсу над одновимірним полягає в тому, що він дозволяє користувачу контролювати

зміну більшої кількості характеристик і у більш чутливий спосіб наближатися до ідеального вибору.

За допомогою фільтрів в одновимірному інтерфейсі користувач може зафіксувати значення деяких характеристик за допомогою фільтрів і рухатися в сторону збільшення чи зменшення осьової характеристики. При цьому усі інші характеристики об'єкта змінюються хаотично і неконтрольовано. Тому, якщо користувачу необхідно збільшити або зменшити неосьову характеристику, йому потрібно або змінити осьову характеристику або перефільтрувати всю вибірку.

На противагу, у багатовимірному інтерфейсі користувач може збільшувати чи зменшувати одну з n осьових характеристик, при цьому залишаючи $n-1$ характеристику, за можливості, незмінними. Тобто, користувач може контрольовано по чергово збільшувати і зменшувати осьові характеристики, наближаючись таким чином до оптимального вибору.

Наступною перевагою багатовимірного інтерфейсу є те, що він значно більш інтерактивний за одновимірний. Багатьом подобається гортати стрічку з цікавими пропозиціями. Багатовимірним інтерфейс дозволяє її гортати в декількох напрямках. Це наближає застосунок, який його імплементує, до гри в пошук найкращого об'єкту.

Також, багатовимірний інтерфейс краще підходить для відображення даних, які мають структуру графа, ніж одновимірний, оскільки його будова зручна для переміщення між вершинам.

Підсумуємо переваги багатовимірного інтерфейсу:

- можливість контрольовано змінювати n осьових характеристик;
- імерсивність;
- зручний для відображення даних, які мають структура графа;

Далі розглянемо недоліки багатовимірного користувацького інтерфейсу у порівнянні з одновимірним.

1.2.1 Недоліки багатовимірного інтерфейсу пошуку

Перш за все, багатовимірний користувацький інтерфейс сильно перегружений порівняно з одновимірним. При використанні багатовимірного інтерфейсу користувач бачить перед собою купу стрілочок, які, можливо, йому зовсім не потрібні. Це може заплутати користувача, що призведе до значних часових втрат. На противагу, у одновимірному інтерфейсі є тільки один вектор руху, і тому розібратися з ним значно простіше.

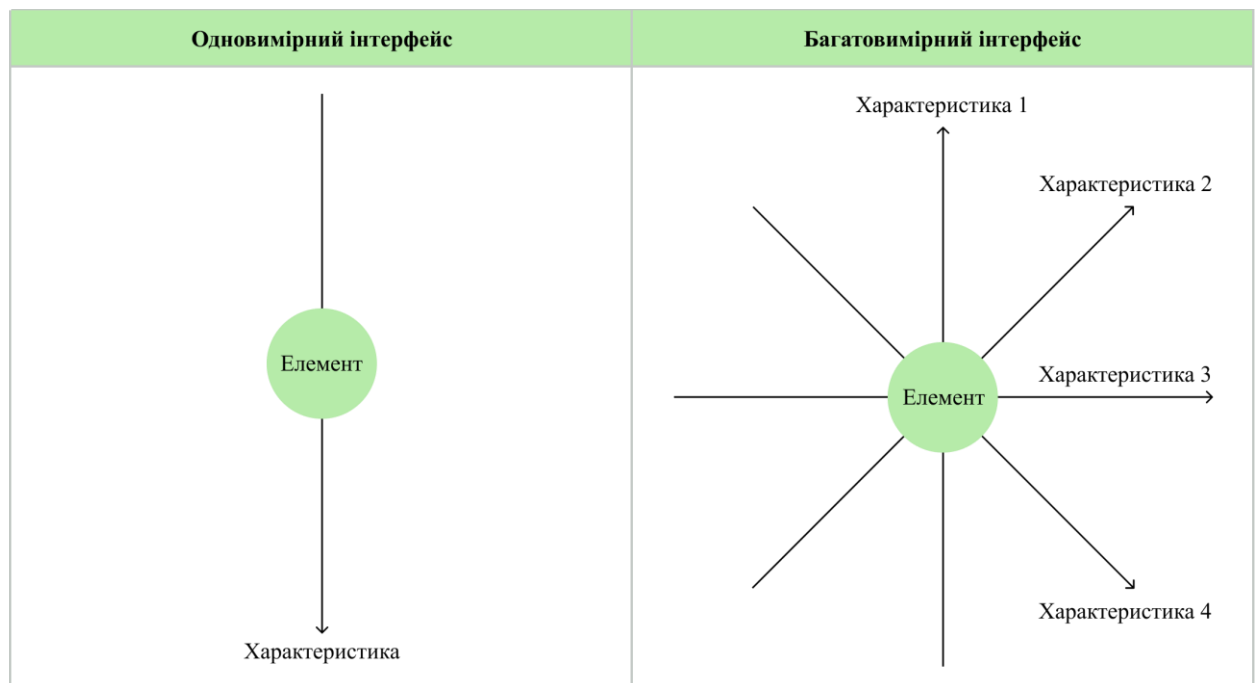


рис. 3. Багатовимірний і одновимірний інтерфейси

Наступним недоліком багатовимірного інтерфейсу перед одновимірним є те, що його функціонал може бути надлишковим. Іншими словами, якщо користувач може визначити точні критерії пошуку, а отже, виставити коректні фільтри і сортування, імерсійний багатовимірний інтерфейс буде тільки заважати.

Окремо від порівняння інтерфейсів слід зазначити обмеженість багатовимірного користувацького інтерфейсу у кількості вимірів. Чим більше осей характеристик, тим складніше, користувачу рухатися за ними. Легко уявити гортання стрічки згори донизу і зліз вправо. Трохи важче це робити по діагоналям. Якщо ж поділити діагоналі навпіл, це стане майже неможливо.

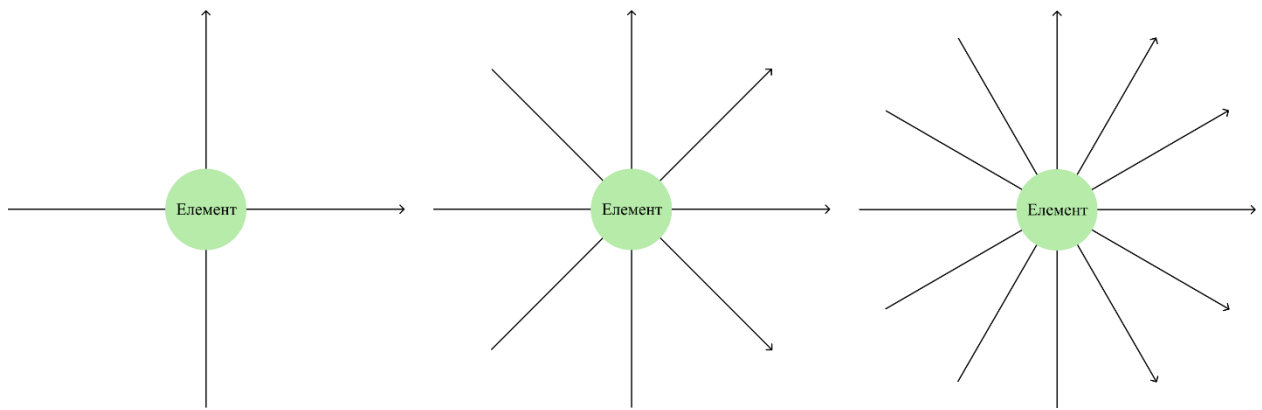


рис. 4. Ускладнення багатовимірного інтерфейсу

Проте, цей недолік можна подолати шляхом створення альтернативних осей характеристик. Таким чином користувач може сам підбирати ті осі, які йому необхідні для пошуку, але це знову ж ускладнює багатовимірний інтерфейс.

Підсумуємо недоліки багатовимірного інтерфейсу:

- перегруженість;
- можлива надлишковість функціоналу;
- обмеженість у кількості вимірів

На основі визначених переваг і недоліків багатовимірного інтерфейсу над одновимірним у наступному параграфі створимо їх порівняльну характеристику.

1.3 Порівняльна характеристика одновимірного і багатовимірного інтерфейсу

Представимо порівняльну характеристику у форматі таблиці

Тип інтерфейсу	Одновимірний	Багатовимірний
Кількість вимірів	1	n, n \geq 2
Дані	будь-які	багатовимірні
Цільова аудиторія	Точно знає значення характеристик об'єкта, який шукає	Або зовсім не має цілі пошуку, або знає тільки тип об'єкту, який шукає і які характеристики в ньому найбільше цікавлять

Додаткові функції	<ul style="list-style-type: none"> - фільтрація; - сортування; 	<ul style="list-style-type: none"> - фільтрація; - сортування; - альтернативні осі характеристик;
Переваги	<ul style="list-style-type: none"> - лаконічність; - простота використання; 	<ul style="list-style-type: none"> - лаконічність; - можливість контролю зміни n осьових характеристик; - імерсивність; - зручність відображення даних, які мають структура графа;
Недоліки	<ul style="list-style-type: none"> - відсутність гнучкості пошуку; - відсутність контролю зміни більше ніж однієї характеристики; 	<ul style="list-style-type: none"> - перегруженість; - можлива надлишковість функціоналу; - обмеженість кількості вимірів;

Провівши порівняльну характеристику одновимірного і багатовимірного користувацького інтерфейсу підведемо підсумок розділу.

1.3 Висновки до розділу 1

Основна ціль пошукових систем - це скоротити час пошуку необхідної інформації або продукту. Заради цього програмісти працюють над покращенням ранжування результатів пошуку, вигадують і реалізують нові способи фільтрації результатів.

Багатовимірний інтерфейс — це один із способів покращення ефективності пошукової системи шляхом відображення багатовимірних даних на площині.

Порівняно з одновимірним інтерфейсом, багатовимірний може як покращити середній час знаходження об'єкту, так і значно його погіршити. Це залежить від структури даних і цільової аудиторії застосунку.

Багатовимірний інтерфейс кращий вибір, ніж одновимірний, якщо:

- дані багатовимірні;
- дані мають структуру графа;
- цільова аудиторія заходить у застосунок без чітко сформованих критеріїв до продукту;

Багатовимірний інтерфейс можна створити із застосуванням багатовимірного адресного сортування, про яке детальніше буде йтися у наступному розділі.

РОЗДІЛ 2: Багатовимірне адресне сортування

У цьому розділі означимо поняття багатовимірне адресне сортування і складемо порівняльну характеристику підходів до його реалізації шляхом визначення переваг і недоліків кожного.

2.1 Поняття багатовимірне адресне сортування

“Під багатовимірним впорядкуванням будемо розуміти сортування за багатьма ознаками чи критеріями та збереження цих результатів таким чином, що від будь-якого елемента сукупності можна швидко (миттєво, без використання пошуку та/або сортування) перейти до інформації щодо іншого елемента, який є наступним чи попереднім за однією з ознак/характеристик, за якою було здійснено сортування. Під виміром будемо розуміти ознаку, характеристику сукупності об’єктів, за якою було здійснено адресне сортування”[2; 3].

Іншими словами, багатовимірне адресне сортування – це сортування адрес даних за багатьма вимірами, яке для зручності користувача вимагає кластеризованого збереження даних, оскільки при отриманні "сусіднього" (наступного/попереднього) елемента по одному виміру користувач може отримати зовсім не схожий на поточний об’єкт пошуку.

В наступному параграфі розглянемо варіанти збереження даних для багатовимірне адресне сортування

2.2 Підходи до збереження даних для багатовимірне адресне сортування

У роботі “Метод кластеризації з використанням багатовимірне адресне сортування”[4] автори розглядають такі варіанти збереження даних, для багатовимірне адресне сортування:

- за допомогою масивів індексації;
- за допомогою двозв’язних списків;
- за допомогою зовнішніх ключів;

Окрім вищезгаданих способів, цікаво розглянути ще збереження даних за допомогою кластеризації бази даних

Розглянемо кожен детальніше.

2.2.1 Збереження даних за допомогою масивів індексації

Ідея полягає у збереженні адрес елементів у n масивах, кожен з яких впорядкований за n -ою характеристикою.

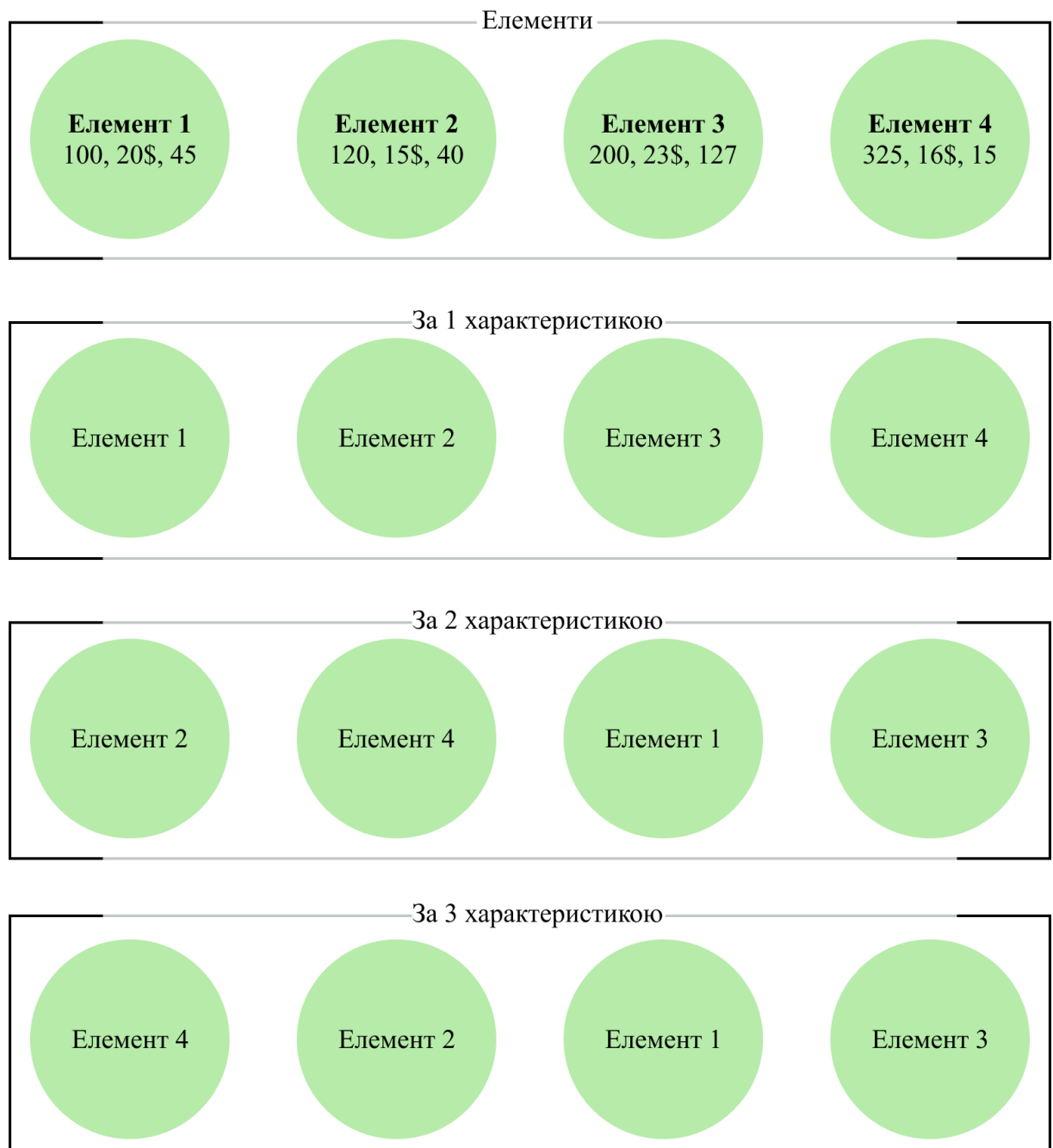


рис. 5. Збереження даних за допомогою масивів індексації

Поглянемо на часову складність елементарних операцій такого підходу.

Часова складність елементарних операцій

Час знаходження наступного елемента у будь-якому напрямі дорівнює часу доступу до елемента масиву за індексом i і рівняється $O(1)$, але при зміні варіанту сортування, потрібно спочатку знайти положення елемента у масиві, тому часова складність зростає до $O(\log n)$.

Час знаходження наступного можна покращити до $O(1)$ за умови створення до кожного масиву адрес елементів оберненого, проте тоді кількість масивів збільшиться удвічі.

Час додавання нового елемента дорівнює часу визначення місця вставки і вставки нової адреси у всі n масивів. Час пошуку місця вставки = $O(n)$, час вставки нового елемента до масиву = $O(n)$. Загалом, час вставки нового елемента = $O(n)$.

Аналогічно, час видалення елемента = $O(n)$

Переваги і недоліки

Перевагами цього методу:

- простота в реалізації;
- низька часова складність знаходження наступного елемента ;
- низька часова складність доступ до елемента за індексом;

Недоліки:

- надмірна дуплікація адрес елементів ;
- висока часова складність вставки і видалення;
- низький рівень безпеки збереження даних;

Далі розглянемо збереження даних за допомогою двозв'язних списків;

2.2.2 Збереження даних за допомогою двозв'язних списків

Ідея полягає в збереженні даних, які мають структуру графа, в незмінному форматі. В даному підході кожен елемент описується вершиною(посилання на елемент) і ребрами(посилання на сусідні вершини).

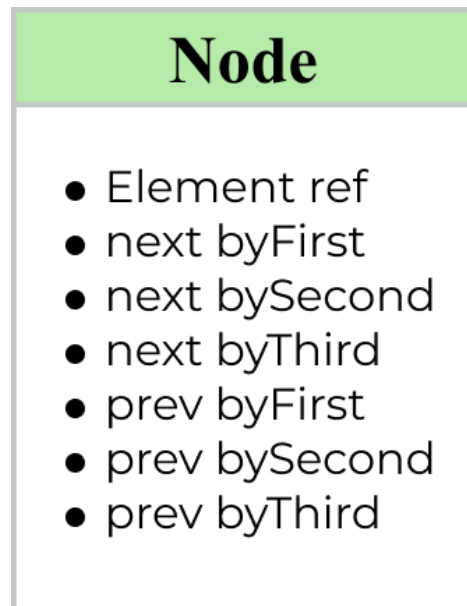


рис. 6. Нода двозв'язного списку

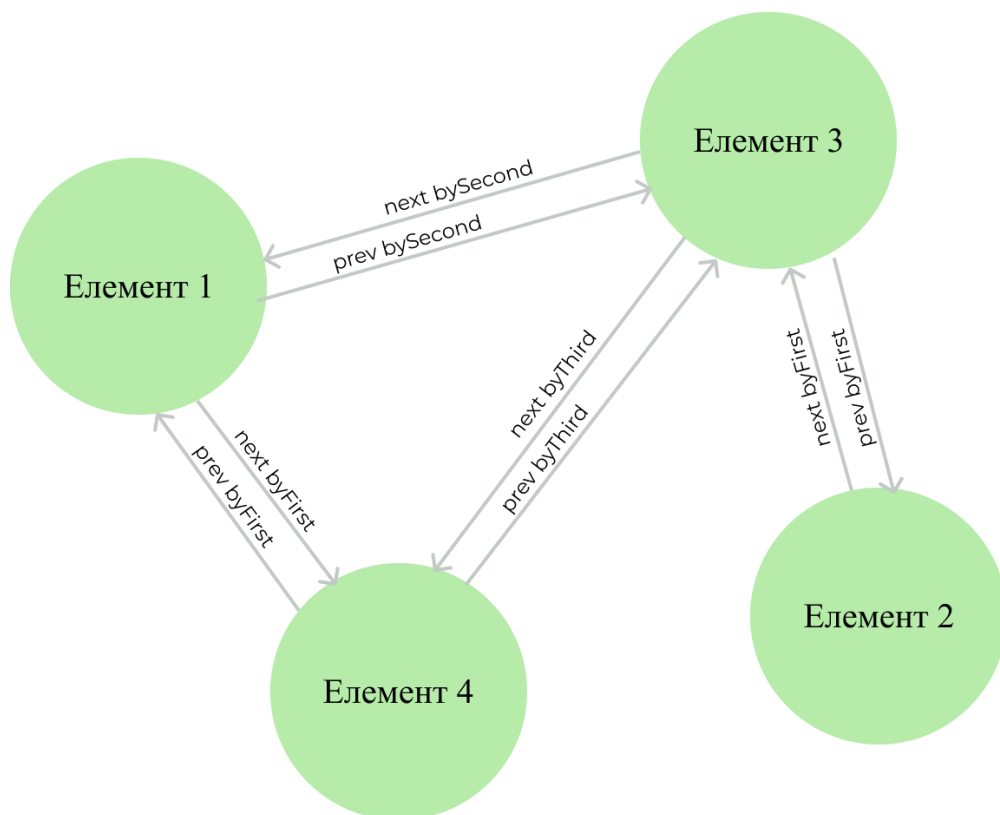


рис. 7. Двозв'язний список

Поглянемо на часову складність елементарних операцій такого підходу.

Часова складність елементарних операцій

Час знаходження наступного елемента у будь-якому напрямі дорівнює $O(1)$.

Часова складність отримання елемента за індексом = $O(n)$.

Час додавання нового елемента, складається з часу пошуку місця нового елемента і часу вставки.

В найгіршому випадку час пошуку місця нового елемента = $O(n)$, час вставки $O(1)$, тому час додавання нового елемента = $O(n)$.

Аналогічно, час видалення елемента = $O(n)$.

Переваги і недоліки

Переваги цього методу:

- простота в реалізації
- низька часова складність знаходження наступного елемента.

Недоліки

- висока часова складність знаходження елемента за індексом;
- висока часова складність вставки і видалення;
- дублювання зв'язків елементів;

Далі розглянемо збереження даних за допомогою зовнішніх ключів.

2.2.3 Збереження даних за допомогою зовнішніх ключів

Ідея даного підходу подібна до попереднього, проте тепер зв'язки з сусідніми елементами реалізуються через зовнішній ключ.

Особливості роботи з базою даних додає переваги і недоліки.

Node	
PK	Element ref
FK	next byFirst
FK	next bySecond
FK	next byThird
FK	prev byFirst
FK	prev bySecond
FK	prev byThird

рис. 8. Схема бази даних

До переваг попередньо параграфу додається:

- доступу до елемента за індексом стане $O(\log n)$;
- високий рівень безпеки збереження даних;

До недоліків додається можливе перевантаження бази даних, оскільки, наприклад, при вставці або видаленні елемента, потрібно змінити усіх його сусідів.

Далі розглянемо збереження даних за допомогою розподілення даних по різних нодам бази даних.

2.2.4 Збереження даних за допомогою кластеризації бази даних

Ідея полягає в тому, щоб утворити кластеризацію таблиці за усіма характеристикам, які в попередньому варіанті були зовнішніми ключами. Таким чином можливо швидко знаходити об'єкти, які знаходяться поруч в одному кластері.

Node	
PK	Element ref
	First Second next Third

рис. 9. Схема бази даних

У цьому варіанті ми найменше контролюємо процес запису і збереження даних. Форма збереження даних, а отже і швидкість читання, запису і видалення, залежить від обраної бази даних.

Розглянувши кожен спосіб збереження даних окремо і вивівши переваги і недоліки, створимо порівняльну характеристику

2.3 Порівняльна характеристика підходів до варіанти збереження даних, для багатовимірного адресного сортування

Представимо порівняльну характеристику у форматі таблиці

Спосіб збереження	За допомогою масивів індексації	За допомогою двозв'язних списків	За допомогою зовнішніх ключів	За допомогою кластеризації бази даних
Часова складність пошуку наступного елемента	$O(1)$	$O(1)$	$O(1)$, залежить від бази даних	$O(1)$, залежить від бази даних
Часова складність пошуку за індексом	$O(1)$	$O(n)$	$O(\log n)$, залежить від бази даних	$O(\log n)$, залежить від бази даних

Часова складність вставки і видалення	$O(n)$	$O(n)$	$O(n)$, залежить від бази даних	$O(\log n)$, залежить від бази даних
Кількість потрібного місця (існує k елементів, кожен займає m біт і має n характеристик)	$2knm$ біт	$k(m+2nm)$ біт	$k(m+2nm)$ біт	$kn(m+nm)$ біт
Безпечність	низька	низька	висока	висока

Провівши порівняльну характеристику підходів до збереження даних для багатовимірного адресного пошуку, підведемо підсумок розділу.

2.4 Висновки до розділу 2

Багатовимірне адресне сортування – це недооцінений спосіб впорядкування і збереження інформації, який дозволяє реалізувати багатовимірний інтерфейс.

Було розглянути чотири підходи до його імплементації, в ході чого визначені переваги і недоліки кожного з них.

Підхід варто обирати з врахуванням таких критеріїв:

- кількість даних, які необхідно зберігати;
- витрати в разі втрати даних;
- вимоги до швидкодії читання;
- вимоги до швидкодії читання доступу за індексом;
- вимоги до швидкодії читання запису і видалення;
- співвідношення між читанням і записом даних;

Як підхід для збереження даних, потрібних для реалізації практичної частини даної курсової роботи було обране збереження за допомогою кластеризації бази даних, оскільки воно видається найцікавішим.

РОЗДІЛ 3: Проектування і програмна реалізація

Завдання даної курсової роботи: спроектувати та розробити WEB застосунок, який використовує багатовимірний інтерфейс для магазину одягу. Такий тип інтерфейсу потребує багатовимірного адресного сортування. В наступному параграфі опишемо основні підходи до реалізації багатовимірного адресного сортування.

3.1 Обґрунтування вибраної предметної області

Згадаймо з 1 розділу критерії оптимальності вибору багатовимірного інтерфейсу:

- дані багатовимірні;
- дані мають структуру графа;
- цільова аудиторія заходить у застосунок без чітко сформованих критеріїв до продукту;

Шопінг задовольняє 2 з трьох критеріїв. По перше, дані про річ багатовимірні, а по друге, частіше за все люди відправляються до магазину з ціллю купити декілька футболок і штани, або ж взагалі просто знайти щось симпатичне. Навіть з означення шопінгу як процесу вибору речей зрозуміло, що людина не має абсолютно точних критеріїв покупки.

Тому, багатовимірний інтерфейс пошуку підходить до вибраної предметної області.

Далі постає питання, що саме буде включати в себе користувацький інтерфейс. На нього дасть відповідь наступний параграф.

3.2 Складові багатовимірного користувацького інтерфейсу

Багатовимірне адресне сортування займає значну площу інтерфейсу, проте, окрім нього реалізовано:

- можливість змінювати осі характеристик;
- можливість змінювати фільтрацію;

Було визначено і імplementовано основні критерії фільтрації, які частіше за все використовують користувачі:

- фільтрація за популярністю;

- фільтрація за вартістю;

Важко переоцінити важливість вибору потрібних характеристик як осьових. Адже це основний інструмент пошуку у застосунку.

Були обрані:

- тон/яскравість;
- тип;
- стиль;
- популярність/вартість;

Скісна риска означає можливість переключення характеристики на осі. Ця функція дозволяє користувачу гнучко налаштувати інтерфейс під себе, для того, щоб зробити пошук ще комфортнішим.

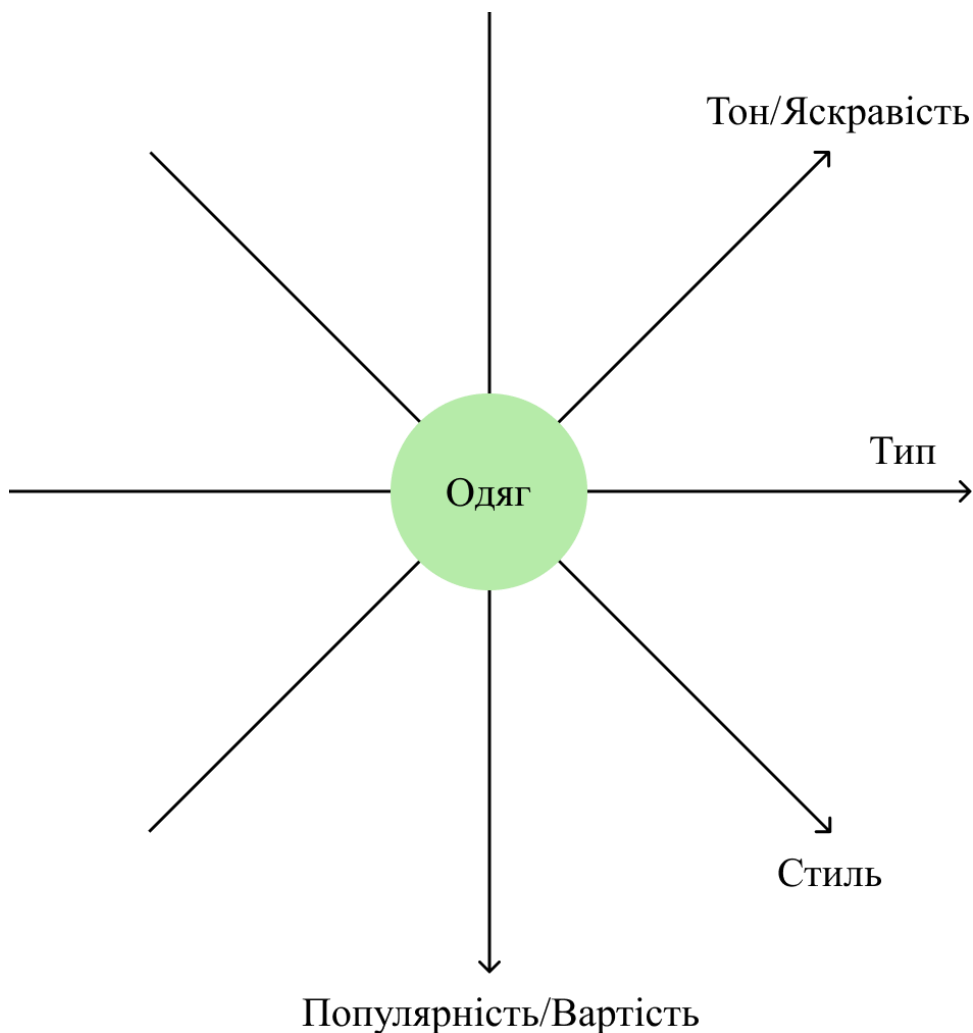


рис. 10. Схематичне відображення осей характеристик

Зібравши все, користувацький інтерфейс має такий вигляд:

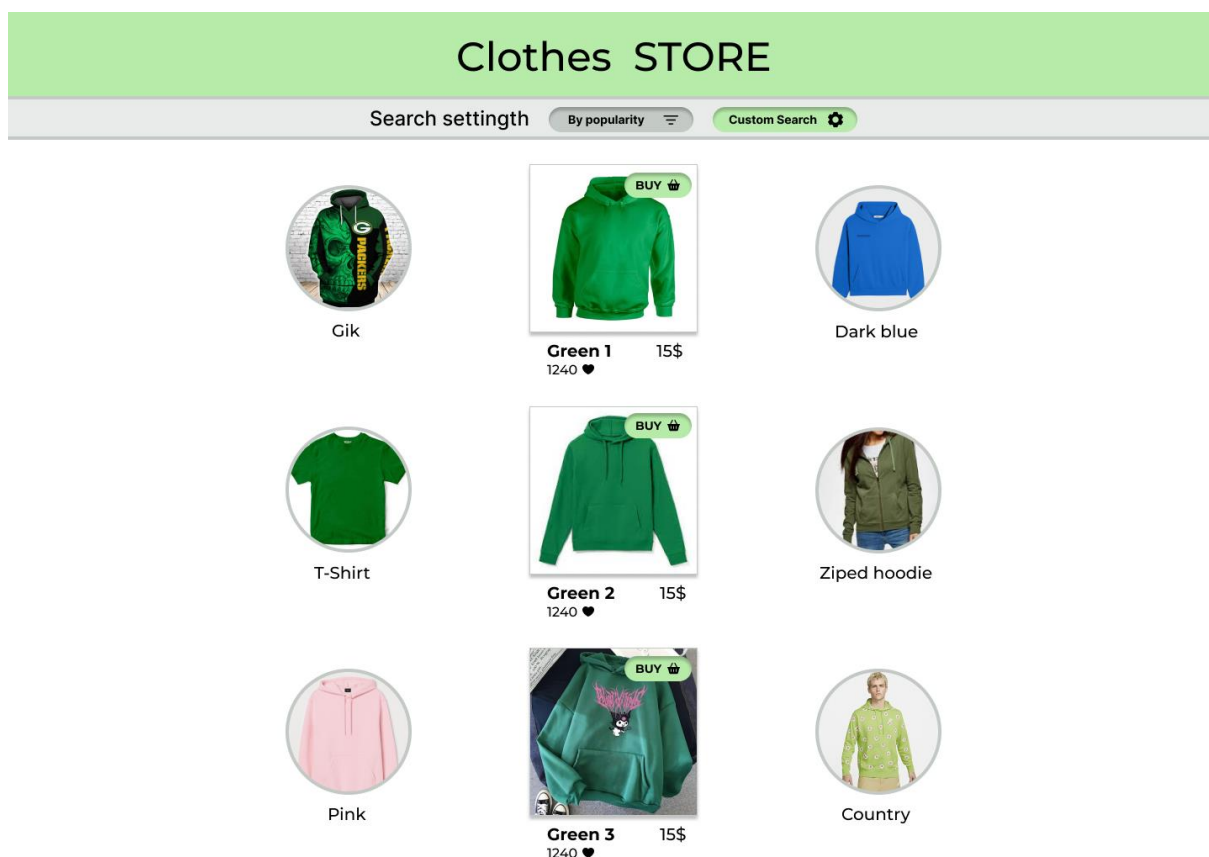


рис. 11. Головна сторінка

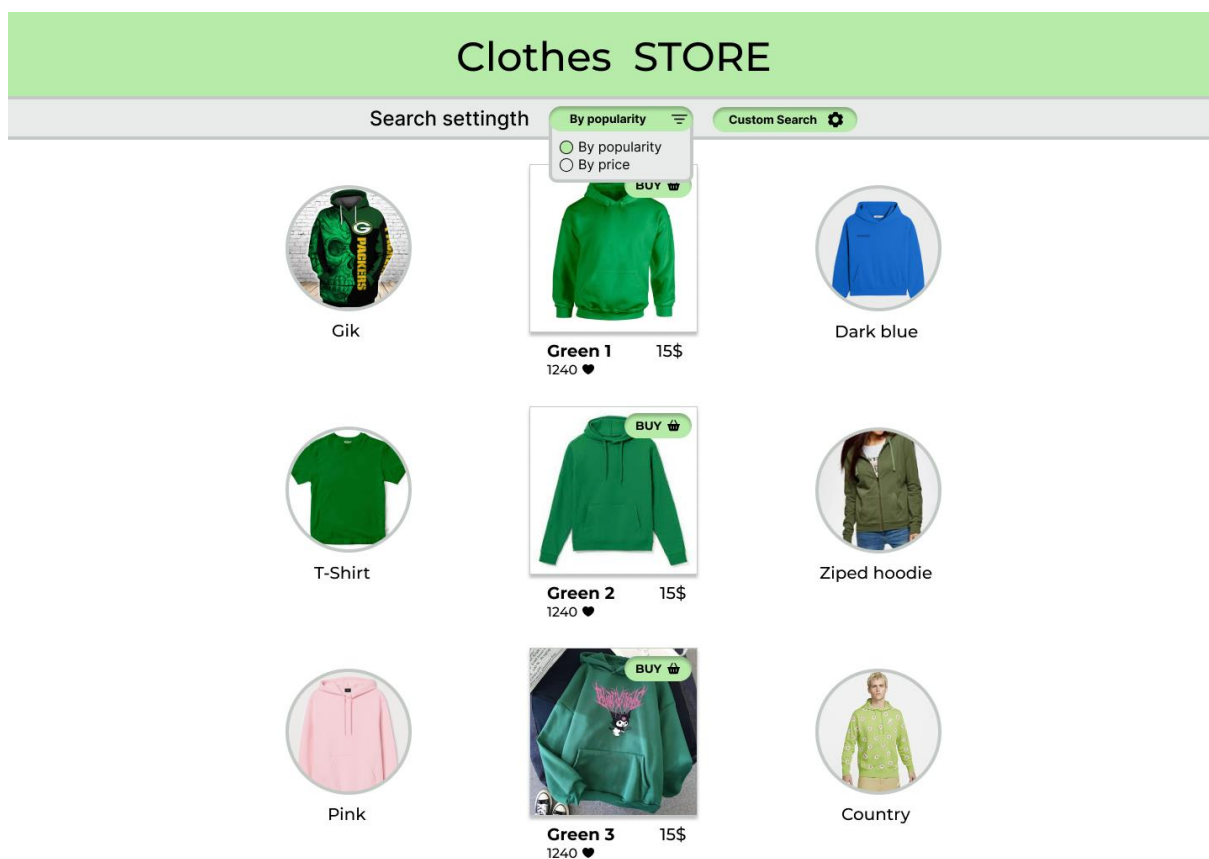


рис. 12. Головна сторінка, вибір фільтру

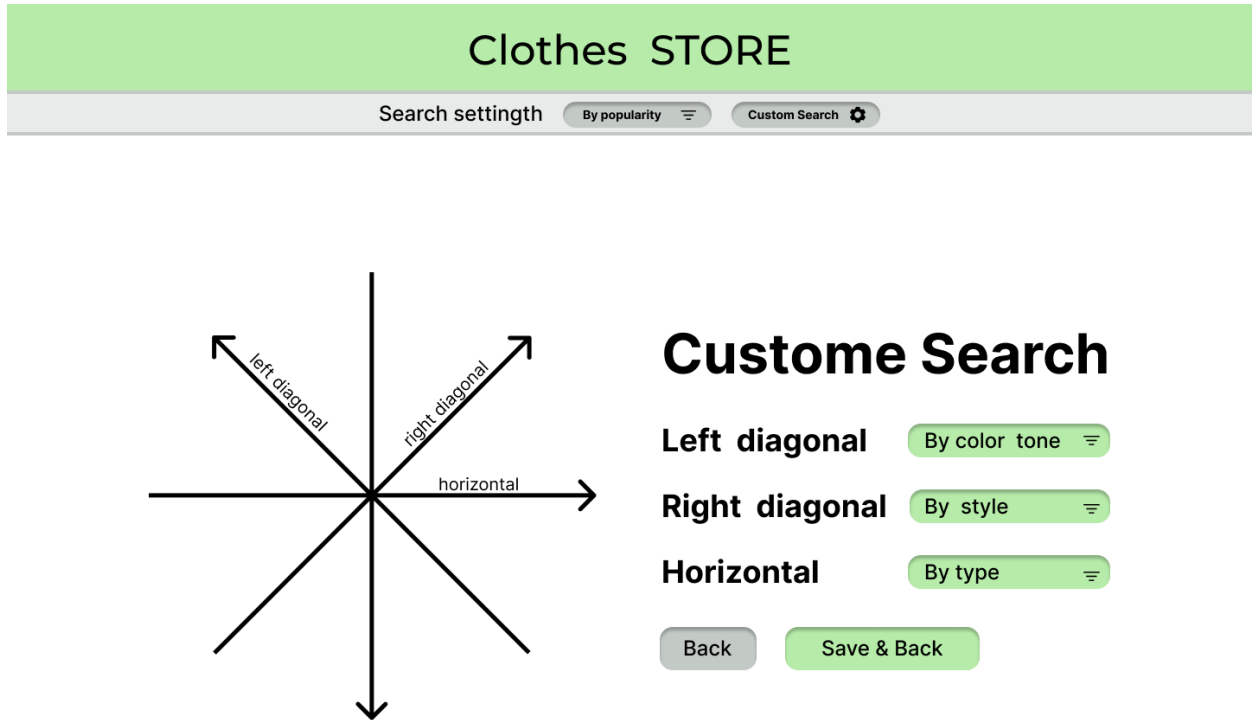


рис. 13. Сторінка вибору осей характеристик

Наступний кроком після візуалізації функціоналу магазину одягу – це вибір стеку технологій.

3.3 Стек технологій

Як мова програмування було обрано Python, оскільки було цікаво попрацювати саме з ним. Інтерфейс реалізований за допомогою бібліотеки flask, яка дозволяє використовувати готові шаблони елементів сторінки і підключати html код для розмітки. Для доступу до Cassandra використовується cassandra-driver, який дозволяє підключитися до серверу на якому хоститься база даних.

Для того, щоб захостити базу даних використовується Docker. “Docker — це відкрита платформа для розробки, доставки та запуску програм”.[5]

Для збереження даних була обрана Apache Cassandra, оскільки вона підходить до поставлених у застосунку завдань.

“Apache Cassandra — це розподілена база даних NoSQL з відкритим вихідним кодом, якій довіряють тисячі компаній за масштабованість і високу доступність без шкоди для продуктивності”.[6] Apache Cassandra відноситься до AP (availability, partition tolerance) баз даних.

У даному випадку дані добре представляються саме NoSQL базою даних, адже не мають реляційних зв’язків між таблицями. Доступність даних значно важливіша за консистентність, адже страшніше, якщо користувач покине сайт через те, що він надто надовго завис, аніж те, що він може пропустити один елемент одягу.

Загалом застосунок використовує дві бази даних. Причини такого вибору описані в наступному розділі.

3.4 Бази даних

Оскільки багатовимірне адресне сортування передбачає роботу з базою даних, яка зберігає адреси елементів, необхідна додаткова база даних, в якій будуть зберігатися елементи(в даному випадку фотографії одягу).

Спершу розглянемо базу для збереження адрес елементів.

3.4.1 База даних для багатовимірного адресного сортування

Як база для збереження даних для багатовимірного інтерфейсу використовується Cassandra. Вона побудована на ідеї розподілення записів між нодами. Розподіл відбувається за Partition Key(ключ партиціювання), в результаті чого дані зберігаються згрупованими за значенням РК. Таким чином запити за значенням РК ефективні, адже замість того, щоб перебирати усі записи в базі, запит відразу знаходить ноду, яка має відповідний РК і шукає результати тільки всередині неї.

Тому, основний підхід до проектування даної бази даних — це орієнтований на запити підхід. Для реалізації даного багатовимірного інтерфейсу необхідна можливість отримувати відповіді на такі запити:

- Взяти наступний елемент за віссю стилю;
- Взяти наступний елемент за віссю типу;
- Взяти наступний елемент за віссю тону;

- Взяти наступний елемент за віссю яскравості;
- Взяти наступний елемент за віссю популярності;
- Взяти наступний елемент за віссю вартості;

Всі ці запити ефективно виконуються, при схемі бази даних, яка зображена на рис. 14.

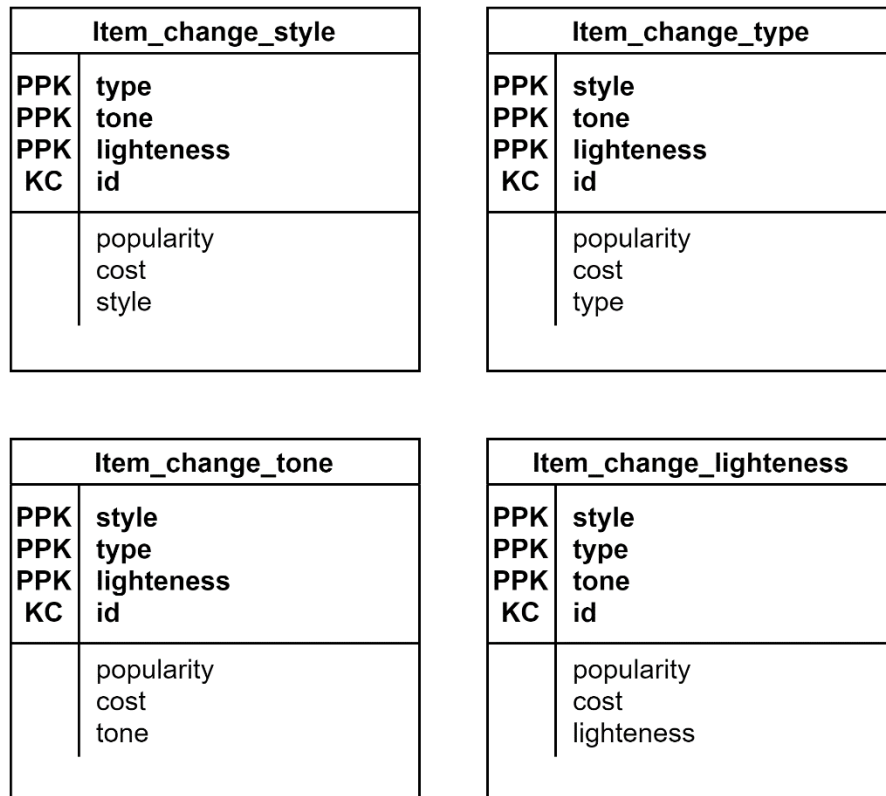


рис. 14. UML бази даних для багатовимірного адресного сортування

Далі розглянемо базу даних для збереження фотографій

3.4.1 База даних для збереження фотографій

Для збереження фотографій підходить будь-яка key-value база даних. Ключ – це ідентифікатор елемента, значенні – фотографія. Оскільки вже використовується Cassandra для збереження адрес елементів, було вирішено використати її для збереження фотографій.

Нижче наведено схему бази даних.

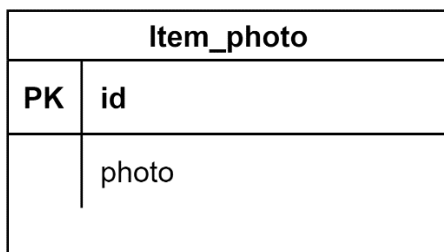


рис. 15. UML бази даних для збереження фотографій

В наступному розділі опишемо спосіб зв'язку застосунку з базами даних.

3.5 Взаємозв'язок застосунку з базами даних

Для коректного відображення елементів одягу, спочатку потрібно отримати дані з бази даних для багатовимірного адресного сортування. Потім з бази, де зберігаються фотографії одягу підтягнути фотографії і відобразити все це на сторінці застосунку. Взаємодія з базами даних відбувається за допомогою CQL запитів.

Нижче наведені Dataflow і Userflow діаграми, які добре ілюструють роботу застосунку.

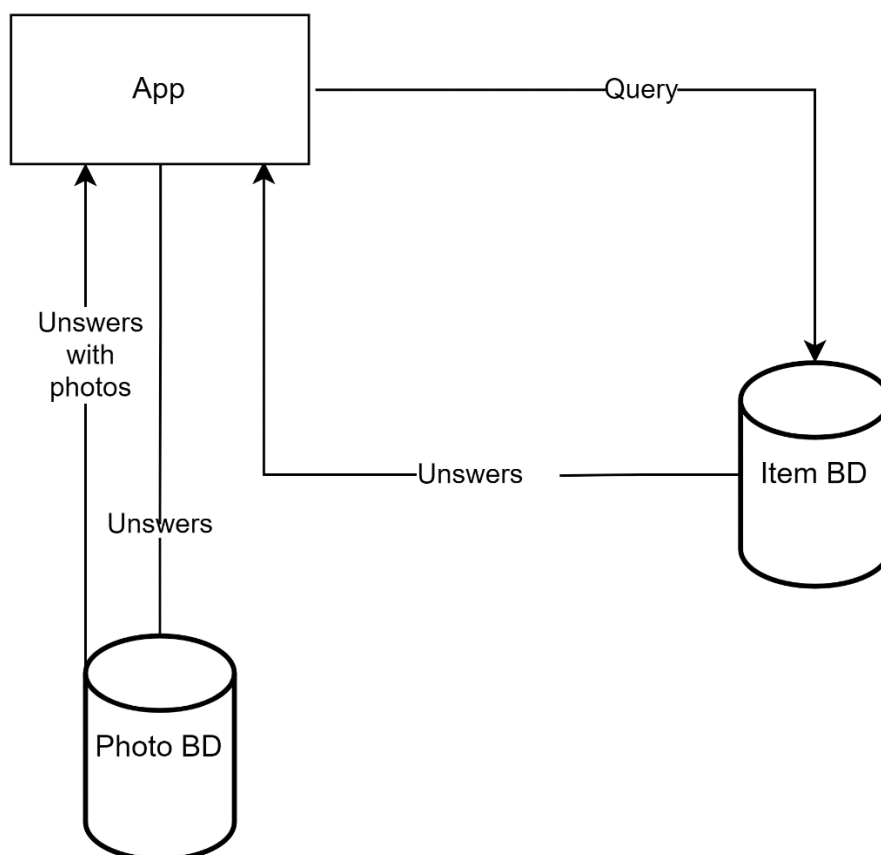


рис. 16. Dataflow діаграма

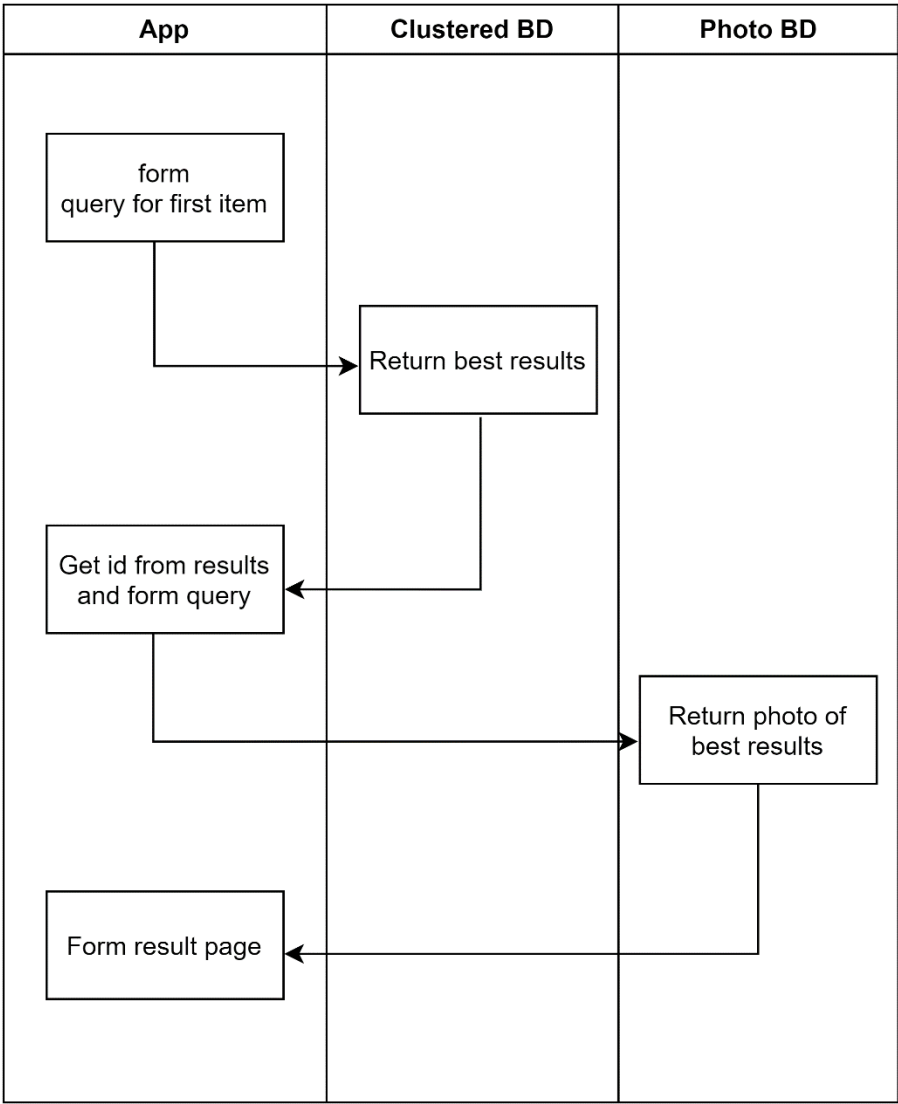


рис. 17. Userflow діаграма

Далі наведемо варіанти для майбутнього покращення застосунку.

3.6 Шляхи покращення застосунку

Одним із варіантів покращення застосунку є додання текстового пошуку. Це дозволить користувачу почати пошук з речі, яка значно ближча до ідеальної.

Крім цього, гарним покращенням стало б розширення системи фільтрації результатів. Наприклад за матеріалом, країною виробництва і т. д.

Ще одним гарним доповненням є додавання нових осьових характеристик. Наприклад, виробник, сезон і т. д.

В наступному параграфі підсумуємо все сказане в цьому розділі.

3.7 Висновки до розділу 3

WEB застосунок, який використовує багатовимірний інтерфейс для магазину одягу, розроблений за допомогою такого стеку технологій:

- Python;
- Apache Cassandra;
- Docker

Реалізовані наступні елементи інтерфейсу:

- багатовимірне сортування;
- можливість змінювати осі характеристик;
- можливість змінювати фільтрацію;

Доступна фільтрація за такими фільтрами:

- популярність;
- ціна;

Доступні такі осі характеристик:

- тон/яскравість;
- тип;
- стиль;
- популярність/вартість;

Для збереження даних застосунку використовується дві бази даних.

Перша - для збереження даних для багатовимірного адресного сортування, друга – для збереження фотографій речей. Взаємодія з базами даних відбувається за допомогою CQL запитів.

Висновки

Головними результатами роботи є:

- створення порівняльної характеристики багатовимірного користувацького інтерфейсу і одновимірного;
- розгляд підходів до реалізації багатовимірного адресного сортування;
- аналіз переваг і недоліків підходів до реалізації багатовимірного адресного сортування;

На поточний момент інформаційні та пошукові системи не використовують багатовимірний інтерфейс заснований на багатовимірному адресному впорядкуванні, хоча переваги над традиційним одновимірним впорядкуванням – очевидні та беззаперечні.

Список використаної літератури

1. TechTarget [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI>
2. Ющенко Ю. О. Багатовимірне впорядкування та його використання для вдосконалення інтерфейсу користувачів інформаційних систем / Ю. О. Ющенко // Наукові записки НаУКМА. Комп’ютерні науки. – 2018. – Т. 1 . – С. 10–13.
3. Ющенко Ю. О. Використання багатовимірного впорядкування для наочного та зручного доступу до інформації / Ю. О. Ющенко // Матеріали XV Міжнародної науково-практичної конференції
4. Крещенко Т. О, Ющенко Ю. О. Метод кластеризації з використанням багатовимірного адресного сортування / Крещенко Т. О, Ю. О. Ющенко // Наукові записки НаУКМА. Комп’ютерні науки. – 2020. – Т. 3 . – С. 83–87.
5. docker docs[Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.docker.com/get-started/overview/>
6. Apache Cassandra[Електронний ресурс] – Режим доступу до ресурсу:
https://cassandra.apache.org/_/index.html