

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Центр післядипломної освіти \_\_\_\_\_

Кафедра \_\_\_\_\_ Програмної інженерії \_\_\_\_\_

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

рівень вищої освіти - другий (магістерський)

Дослідження алгоритмів оптимізації пошуку туристичних подорожей в умовах  
карантину

Виконав:

Студент 2 курсу, групи ПЗЗдм-19-1

Кирєєва І.О.

Спеціальність 121-Інженерія програмного забезпечення

Тип програми освітньо-наукова

Керівник доцент Турута О.П.

Допускається до захисту

Зав. Кафедри, проф. \_\_\_\_\_

З.В. Дудар

2021 р.

## Харківський національний університет радіоелектроніки

Факультет Центр післядипломної освіти  
(повна назва)

Кафедра Програмної інженерії  
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – інженерія програмного забезпечення  
(код і повна назва спеціальності)

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

Освітня програма Інженерія програмного забезпечення  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2021р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Кирєєвої Ірини Олександрівни  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження алгоритмів оптимізації пошуку туристичних подорожей в умовах карантину  
затверджена наказом університету від 26.03.2021 № 34 Стз
2. Термін подання роботи до екзаменаційної комісії 14 травня 2021р.
3. Вихідні дані до роботи методи вирішення задачі багатокритеріальної оптимізації, методи нормалізації критеріїв та пріоритетів, OS Windows, об'єктно-орієнтована мова програмування C#, фреймворк Angular 9, середовище розробки Visual Studio Professional 2019
4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, аналіз методів вирішення багатокритеріальних задач, аналіз методів нормалізації, розробка алгоритму ранжування, проектування програмного додатку.
5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій мета завдання, обґрунтування доцільності розроблення, постановка задачі, аналіз аналогів, аналіз методів вирішення багатокритеріальних задач, проектування системи, розробка алгоритму ранжування, демонстраційні матеріали, можливі удосконалення, висновки.

## 6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	доц. Турута О.П.		

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	18 лютого 2021р.	виконано
2.	Аналіз методів вирішення багатокритеріальних задач	10 березня 2021р.	виконано
3.	Проектування додатку	17 березня 2021р.	виконано
4.	Проектування алгоритму оптимізації	31 березня 2021р.	виконано
5.	Розробка демонстраційного додатку	19 квітня 2021р.	виконано
6.	Підготовка пояснювальної записки	06 травня 2021р.	виконано
7.	Підготовка презентації та доповіді	06 травня 2021р.	виконано
8.	Перевірка на академічний плагіат	07 травня 2021р.	виконано
9.	Нормоконтроль	07 травня 2021р.	виконано
10.	Рецензування	12 травня 2021р.	виконано
11.	Попередній захист	14 травня 2021р.	виконано
12.	Занесення диплома в електронний архів	14 травня 2021р.	виконано
13.	Допуск до захисту у зав. кафедри	14 травня 2021р.	виконано

Дата видачі завдання 25 січня 2021р.Студент \_\_\_\_\_  
(підпис)Керівник роботи \_\_\_\_\_ доцент Турута О.П.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 68 с., 11 рис., 4 табл., 21 джер.

АЛЬТЕРНАТИВА, БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, КРИТЕРІЙ, МАТЕМАТИЧНА МОДЕЛЬ, ПРИЙНЯТТЯ РІШЕННЯ, ТУРИСТИЧНА ПОДОРОЖ.

Об'єктом дослідження є задача прийняття рішення на основі багатокритеріального вибору для обирання туристичної мандрівки в умовах карантину.

Метою роботи є дослідження методів вирішення задачі багатокритеріального вибору на основі вибору туристичної мандрівки, дослідження методів нормалізації критеріїв та розробка програмного додатку для ранжування туристичних подорожей.

Методи розробки базуються на таких технологіях, як фреймворк .NET, C#, Angular 9.

В результаті роботи було проаналізовано методи пошуку кращої альтернативи за заданими критеріями, методи нормалізації для критеріїв та пріоритетів, а також спроектовано систему для ранжування туристичних подорожей, яка має клієнт-серверну архітектуру та спілкується з сторонніми додатками для отримання інформації про подорожі за допомогою архітектури REST.

ALTERNATIVE, CLIENT-SERVER ARCHITECTURE, CRITERION, DECISION MAKING, MATHEMATICAL MODEL, MULTICRITERIAL OPTIMIZATION, TOURIST TRAVEL.

The object of the study is the task of making a decision based on a multi-criteria choice for choosing a tourist trip in quarantine.

The aim of the work is to study the methods of solving the problem of multicriteria choice based on tourist travel, the study of methods of normalization of criteria and the development of a software application for ranking tourist travel.

Development methods are based on such technologies as .NET Framework, C#, Angular 9.

As a result, the methods of finding the best alternative according to the given criteria, normalization methods for criteria and priorities were analyzed, as well as a system for ranking tourist trips was developed, which has a client-server architecture and communicates with third-party applications to obtain travel information using REST architecture.

Я, Кирєєва Ірина Олександрівна, студент гр. ПЗЗдм-19-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження алгоритмів оптимізації пошуку туристичних подорожей в умовах карантину», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомена з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі та постановка задачі.....	9
1.1 Виявлення проблем та актуалізація рішень .....	9
1.2 Проблема прийняття рішення.....	14
1.3 Постановка задачі.....	16
2 Аналіз методів вирішення багатокритеріальних задач вибору .....	18
2.1 Метод головного критерію.....	19
2.2 Методи згортки.....	20
2.2.1 Методи адитивної згортки .....	21
2.2.2 Метод мультиплікативної згортки .....	22
2.3 Лексикографічна оптимізація .....	23
2.4 Метод послідовних поступок.....	24
2.5 Побудова множини Парето .....	25
2.6 Метод «ідеальної» точки .....	25
2.7 Метод справедливого компромісу .....	26
2.8 Результати аналізу методів .....	27
3 Проектування програмної системи.....	29
3.1 UML-проектування системи .....	30
3.2 Проектування архітектури програмної системи .....	32
3.3 Розробка алгоритму ранжування альтернатив .....	33
3.3.1 Нормалізація критеріїв .....	34
3.3.2 Визначення вагових коефіцієнтів.....	37
3.3.3 Використання адитивної згортки .....	38
4 Опис прийнятих програмних рішень .....	41
4.1 Опис засобів розробки .....	41

4.2 Опис програмної системи.....	42
Висновки .....	45
Перелік джерел посилання .....	46
Додаток А Перелік джерел посилання кафедри програмної інженерії .....	48
Додаток Б Звіт результатів перевірки роботи на унікальність тексту .....	49
Додаток В Слайди презентації .....	50
Додаток Г Апробація результатів роботи .....	61
Додаток Д Експертний висновок результатів нормоконтролю.....	68

## ВСТУП

У повсякденному житті комп'ютерні технології є невід'ємною частиною нашого життя. Вони проникли і міцно закріпилися у всіх сферах людської діяльності, що посприяло значному збільшенню обсягу збережених даних. Будь-яку інформацію можна легко відшукати за допомогою мережі Інтернет і мета може бути найрізноманітніша, починаючи від пошуку улюбленого фільму до повного планування подорожі.

На сьогоднішній день все більше людей захоплюються туризмом. Людям подобається відвідувати нові місця, розглядати історичні пам'ятки, архітектуру, знайомитися з культурою різних міст та країн.

Зараз можна знайти безліч ресурсів, що можуть допомогти замовити квитки на транспорт, забронювати готель, орендувати автомобіль на час перебування в іншому місті та інше. Однак, велика кількість людей, не люблять витратити багато часу на формальні питання під час планування і бажають придбати готову путівку, де вони будуть лише насолоджуватись відпочинком. Для цього їм на допомогу приходять веб-ресурси з безліччю мандрівок у різні сторони світу, де завдяки вибору параметрів свого унікального відпочинку, кожен користувач має можливість знайти те, що йому по душі. Але у більшості випадків це не один або два критерії, які допоможуть зробити подорож незабутньою і у цьому криється головна проблема. На людину та її остаточний вибір може впливати обов'язковий перетин великого набору критеріїв або навпаки – виняток, що ускладнює підбір варіацій системою з максимальним влученням у вимоги користувача. Ще тяжчим це стало після пандемії коронавірусу, де певні напрями закриті для туристів з тої чи іншої держави.

Отже, спостерігаємо потребу у системі, яка допоможе на основі багатьох критеріїв різної пріоритетності пропонувати альтернативи відпочинку як змога точніше.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Виявлення проблем та актуалізація рішень

Під час проведення аналізу аналогів та продуктів, маючих подібний функціонал, було виявлено декілька великих продуктів, які користуються попитом. За для того, щоб програмний продукт, розроблений в рамках цього дослідження, мав максимальну влучність в потреби користувача, проаналізуємо плюси та мінуси цих додатків та ймовірні рішення, які вони використовують для рішення своєї задачі. На підставі цього, матимемо можливість зрозуміти, якими якостями та функціональними потребами повинен відповідати розроблений програмний продукт.

Expedia (<https://www.expedia.com/>) [1] – це туристичне онлайн-агенція, яка дозволяє бронювати авіаквитки, готелі, туристичні мандрівки, екскурсії, круїзні лайнери та орендувати автомобілі (див. рис. 1.1). Найпопулярніший веб-сайт на території США та має локальні сайти ще в 30 країнах світу, але на українському та російському ринку ще не був презентований.

Оскільки компанія є однією з лідируючих в своїй галузі, вона дозволяє без проблем спланувати різні варіанти подорожей. Не настільки важливо, чи буде це екстремальна поїздка або ж легкий романтичний вояж, на сайті представлять необхідну інформацію. Відомості про цікавить пункті можна отримати не тільки на веб-ресурсі, але і в окремих мобільних додатках.

Підбір нових пропозицій щодо бронювання готелів, машин, квитків на авіаперельоти та інших послуг тут не припиняється ніколи. На даний момент список відповідних для туристичних і ділових поїздок варіантів уже поповнився майже 155 тис. готелів, що розташувалися в різних точках світу. Додаток викупає набори номерів в багатьох країнах світу, що дозволяє пропонувати найнижчі ціни на ринці.

Якщо ж говорити про попередньому замовленні квитків на літаки, лайнери та інші види транспорту, то в даному випадку перелік компаній сягає 300 найменувань.

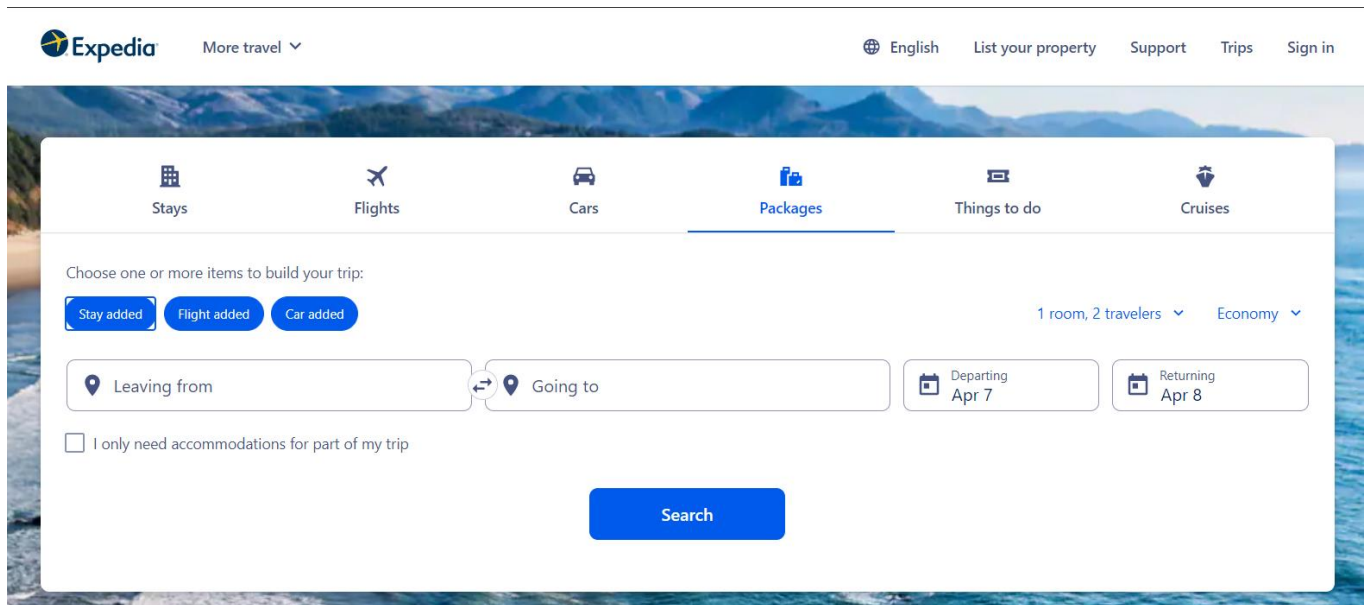


Рисунок 1.1 – Головна сторінка Expedia

Розвиток Expedia не обмежується розвитком тільки одного сайту. Все більшою популярністю користуються однойменні мобільні додатки, які функціонують на базі Android і iOS. На особливу увагу заслуговує нова програма, розроблена спеціально для планшетів. Зручний інтерфейс дозволяє дуже швидко розібратися з керуванням. Втім, цей фактор не є основним. Особлива функціональність полягає в тому, що всі готелі і послуги тут згруповані за окремими категоріями. Так, наприклад, можна придивитися ближче до розділу під назвою «Величні замки». Або «Європейська історія». Представлене різноманітність варіантів практично не знає кордонів.

Booking (<https://www.booking.com/>) [2] – найбільший європейський портал бронювання, що пропонує 247 тисяч готелів в 177 країнах світу (див. рис. 1.2). Веб-сайт перекладено на 41 мову, включаючи російську та українську, а також надає службу підтримки в багатьох країнах світу.

Одною з основних фішок даного додатку – це відсутність плати за бронювання, а саме сайт не бере комісію або фіксовану ставку від користувача, який бронює номер, а отримує відсоток від готелю. Також надає одну з найбільших баз відгуків на готелі по всьому світу, що дозволяє підібрати готель під себе завдяки реальним людям, які залишили свою думку.

**Booking.com** UAN ? Зареєструвати своє помешкання Зареєструватися Увійти

Помешкання Авіаквитки Оренда авто Дозвілля Таксі з/до аеропорту

**Коронавірус (COVID-19): підтримка**

**Знаходьте пропозиції готелів, приватних помешкань та багато іншого...**  
Від затишних замських будиночків до цікавих апартаментів у місті

Куди ви вирушаєте? Заїзд – Виїзд 2 дорослих · 0 дітей · 1 номер Шукати

Я подорожую у справах

**Вигідний початок року**  
Економте від 20% на наступній поїздки – хай 2021 рік почнеться вдало.  
[Знайти пропозиції](#)

**Зекономте від 15%**  
Знайдіть до 30.09 пропозиції на поїздки неподалік або далі від дому.  
[Переглянути](#)

**Найпопулярніші пляжні напрямки**  
Популярно серед мандрівників із вашого регіону.

**Іспанія** **Україна** **Італія** **Таїланд**

Рисунок 1.2 – Головна сторінка Booking

Додаток має чотири етапи пошуку оптимальних варіантів для користувача: вибір по атрибутам(відсутність або присутність чого-небудь), варіації поселення,

якщо не існує кімнат на таку кількість людей, пошук готелю по доступності(цін, відстані) і ранжування – результати, якого відображено на рисунку 1.3.

**Найти**  
Место/название объекта:  
🔍 Киев

Дата заезда  
📅 суббота, 1 мая 2021

Дата отъезда  
📅 воскресенье, 9 мая 2...

8 ночей  
2 взрослых

Без детей 1 номер

Я путешествую по работе

**Найти**

Сохраняйте свободу действий с бесплатной отменой. [Подробнее](#)  
Показать только варианты с бесплатной отменой

**Выбрать по критериям:**

**Ваш бюджет (за ночь)**  
 Установить свой бюджет

<input type="checkbox"/> UAH 0 — UAH 1 640	1185
<input type="checkbox"/> UAH 1 640 — UAH 3 290	471
<input type="checkbox"/> UAH 3 290 — UAH 4 940	131
<input type="checkbox"/> UAH 4 940 — UAH 6 590	36
<input type="checkbox"/> UAH 6 590 +	29

**Киев: найден 1 521 вариант**

Мы рекомендуем для долгой поездки | Дома и апартаменты целиком

**BonApart Hotel** ★★★★★  
Печерский, Киев - Показать на карте - 4 км от центра  
Еще 1 человек просматривал на ваши даты за последние 10 минут

Подходит для долгих поездок  
**Ограниченное предложение**

Апартаменты-студио — 2  
Разные типы кроватей  
**БЕСПЛАТНАЯ отмена бронирования**  
• Без предоплаты  
Вы сможете отменить позже, поэтому успеете забронировать сегодня по этой отличной цене.

8 ночей, 2 взрослых  
UAH 14 520 ~~UAH 26 343~~  
+ налоги и сборы (UAH 384)

Потрясающе 8,9  
952 отзыва  
Комфорт 9,1

**Посмотреть наличие мест**

**Sky Loft Hotel Kyiv by Rixwell** ★★★★★  
Печерский, Киев - Показать на карте - 1,2 км от центра  
- Рядом с метро  
Забронировано 2 раза на ваши даты за последние 24 часа

Двухместный номер с кроватью размера "king-size" — 2  
1 очень большая двуспальная кровать  
**БЕСПЛАТНАЯ отмена бронирования**  
• Без предоплаты  
Вы сможете отменить позже, поэтому успеете забронировать сегодня по этой отличной цене.

8 ночей, 2 взрослых  
UAH 14 968  
+ налоги и сборы (UAH 384)

Потрясающе 8,9  
2 860 отзывов  
Комфорт 9,2

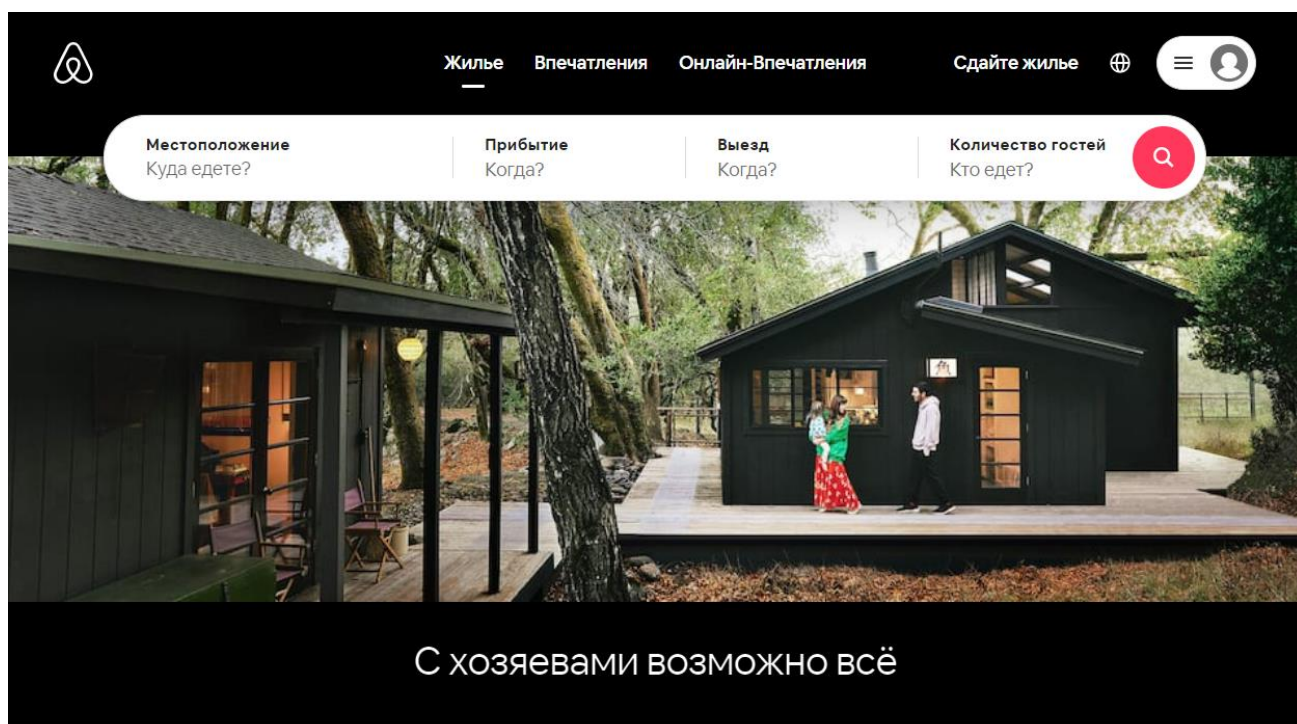
**Посмотреть наличие мест**

**Гостиница Крещатик** ★★★★★  
Потрясающе 8,6  
2 268 отзывов

Рисунок 1.3 – Результат пошуку та ранжування на Booking

Airbnb (<https://www.airbnb.ru/>) [3] – інтернет-ресурс, який допомагає орендувати житло від приватних осіб по всьому світу (див. рис. 1.4). Тобто люди, у яких є зайва кімната або квартира або будинок і бажання спілкуватися з туристами / заробити гроші, діляться всілякими місцями для ночівлі та довготривалого

проживання. Це можуть бути загальні і окремі кімнати, прості квартири, і шикарні апартаменти, і вдома, аж до замків, і навіть намети та будинки на колесах.



## Жилье поблизости



Киев  
7.5 часов езды



Запорожье  
5 часов езды



Полтава  
3 часа езды



Днепро  
3.5 часа езды



Николаев  
8.5 часов езды



Софиевская Борщаговка  
7.5 часов езды

Рисунок 1.4 – Головна сторінка Airbnb

Airbnb має можливість впорядкувати результат пошуку, скориставшись фільтрами, як базові: цікавить нас квартира, кімната або дім, встановити діапазон цін, так і декілька нестандартними фільтрами – зручність, тип житла, мова господаря та інше.

Проте, усі ці додатки мають один мінус, вдосконалення, якого і є метою цієї роботи – після того, як користувач обрав фільтри, він все одно повинен переглянути більшість з підходящих варіантів для того, щоб обрати саме те, що він шукає і максимально вдовольнити свою потребу. Додаток, який має бути розроблено у ході цієї роботи має можливість надавати користувачам обрати пріоритетність критеріїв на основі якого і буде робитися ранжування даних, що допоможе знайти потрібний користувачів варіант на вершині пошуку.

## 1.2 Проблема прийняття рішення

Практично, кожен день людина стикається з ситуаціями, де існує якась множина можливостей і людина повинна обрати одну з них, яка на її погляд найбільш влучна. В житті кожної людини прийняття рішення є одною з найважливіших складових, яка визначає майбутнє, а перекладаючи на формальну мову – усе життя це послідовність рішень. Роблячи будь-який вибір у своєму житті, людина прагне передбачити усі можливі наслідки у разі того чи іншого вибору, що інколи просто неможливо; зіставити переваги та недоліки усіх варіантів, котрих можуть бути тисячі або десятки тисяч. Усе це робить кожен вибір неоднозначним, адже обираючи можливо лише припустити результат.

В теперішній час, техніка може обчислювати терабайти даних, виконуючі складні операції, проте, людина має вміння відібрати головні фактори, що є критичними у даному конкретному випадку особливо для неї, й неважливі, що можуть бути спрощені; також особливістю людського мозку є доповнювати невизначену інформацію своїми припущеннями, що дає можливість порівнювати непорівнювальні альтернативи. Однак, жоден мозок не в змозі обробити сотні тисяч

альтернатив, що нашоухує на потребу засобу, що полегшить людині прийняти рішення базуючись на множині критеріїв.

Теорія прийняття рішень вивчає задачі найкращого вибору, що дозволяє здійснювати вибір більш ефективно, використовуючи усю існуючу інформацію про вподобання людини. Непомірно широкий та особливо важливий з практичної точки зору, у цій області дослідження, – це клас багатокритеріальних задач, в яких якість прийнятих рішень оцінюється по кільком критеріям одночасно [4]. Головною інформацією в такого роду задачах є інформація про вподобання особи, яка приймає рішення, а саме пріоритетність важливості критеріїв.

Прийняття рішення завжди передбачає вибір одного з можливих варіантів дій, які прийнято називати альтернативами. І коли альтернатив багато (сотні або тисячі), людина, яка приймає рішення потребує засоби допомоги у прийнятті рішень, бо не може проаналізувати їх усі.

Треба зазначити, що процес прийняття рішення не є одномоментною задачею, а складається з трьох основних фаз: побудова множини альтернатив, аналіз характеристик альтернати на основі заданих критеріїв та вибір найкращої альтернативи (див. рис. 1.5).

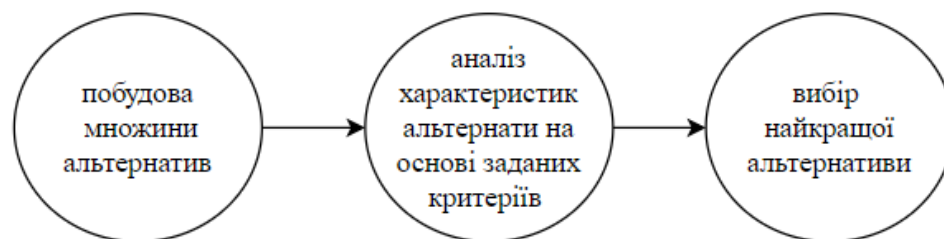


Рисунок 1.5 – Схема процедури прийняття рішень

В більшості випадків розпізнають три основні типи задач прийняття рішень:  
 – сортування або упорядкування альтернатив. Суть такого типу задач досить прозора – визначити порядок підходящих альтернатив. Яскравим прикладом

служить список справ на день, які потрібно зробити від найважливішого до дрібниць, які не мають чітких дедлайнів;

– розподілення альтернатив за класами рішень. Цей тип задач зустрічається найчастіше, коли особа повинна згрупувати набори альтернатив за якимось критерієм, що допоможе у майбутньому.

– вибір найкращої альтернативи. Вважається одним з головних типів задач в дисципліні прийняття рішень. Вибір міста роботи, навчання або покупка ноутбука, мандрівки добре характеризують такий тип задач.

Удосконалення багатокритеріальних методів з середини 90-х років ХХ століття стимулюється розвитком комп'ютерних мереж, зажадали розробити такі методи підтримки прийняття рішень, які можуть бути використані в мережі Інтернет, а також в інших глобальних і локальних мережах.

В даній кваліфікаційній роботі предметною галуззю є оптимізація пошуку туристичних подорожей під час карантину на підставі різноманітних критеріїв та їх пріоритетності.

### 1.3 Постановка задачі

Базуючись на аналізі предметної галузі, а саме теорії прийняття рішень у багатокритеріальних задачах, та аналогів з наданнях подібних послуг, які потребують вибір людини на підставі множини критеріїв, маємо змогу зробити такі висновки:

– більшість ресурсів, які надають можливість придбання туристичної подорожі, не надають можливість придбати її без телефонного дзвінка та розмови з туристичним агентом. В наш час, це є велика проблема, адже все більше людей

використають месенджери і віддають перевагу повідомленням замість телефонних дзвінків;

– більшість кількості сайтів для купівлі мандрівки не передбачають увесь перелік мандрівок, а лише найвідоміші або ті, що потрібно продати, а повний перелік можна знайти лише у туристичного оператора, зробивши той самий дзвінок.

Беручи до уваги усі недоліки та переваги, які були визначені у існуючих аналогів або ресурсів маючих подібний функціонал, маємо можливість визнати актуальність роботи, яка полягає у потребі створити додаток, головною особливістю якого буде можливість підбору ідеальної туристичної мандрівки, на підставі визначення її критеріїв, обробка яких надає можливість визначити мету роботи.

Метою роботи є аналіз методів рішення багатокритеріальних задач, на підставі якого буде визначено підходящий для даної предметної галузі, а саме вибір туристичної мандрівки, розробка методу ранжування альтернатив з використанням математичної моделі, на основі вибраних методів та програмна імплементація математичної моделі для підтвердження результатів дослідження. Програмну систему необхідно реалізувати у вигляді веб-системи, що матиме серверну частину та веб-додаток, які будуть незалежні один від одного, а отже мати можливість бути заміненіми на інші у разі потреби.

Серверна частина має реалізовувати бізнес-логіку застосунку, виконувати збереження та вибірку даних із бази даних, виконувати обробку даних, реалізовувати інтерфейс взаємодії у вигляді REST JSON API, бути простою у розгортанні та масштабуванні [5].

Клієнтський застосунок має надавати користувачеві привабливий та простий графічний інтерфейс для взаємодії із системою, виконувати запити до сервера через REST JSON API відповідно до потреб користувача, бути сумісним із сучасними веб-браузерами.

## 2 АНАЛІЗ МЕТОДІВ ВИРІШЕННЯ БАГАТОКРИТЕРІАЛЬНИХ ЗАДАЧ ВИБОРУ

Задачі багатокритеріального вибору виникають у тих випадках, коли існує декілька цілей, котрі не можуть бути відібрані одним критерієм (наприклад, вартість мандрівки та кількість людей). Для цього потребується знайти точку області допустимих рішень, котра мінімізує або максимізує усі ці критерії – що допоможе підібрати найкраще рішення для користувача [6].

Задача багатокритеріального вибору має наступну постановку: нехай задано множину допустимих рішень, яку визначимо, як  $D$  і множину критеріїв  $f_1(x), f_2(x), \dots, f_m(x)$ , визначених на усіх елементах множини  $D$ :

$$F(x) = (f_1(x), \dots, f_m(x)) \rightarrow \max, x \in D, \quad (2.1)$$

де  $F(x)$  –  $m$ -вектор-функція аргументу  $x \in D$ .

Потребується знайти рішення  $x^* \in D$  для котрого значення критеріїв максимально можливі.

Якщо функції  $f_1(x), f_2(x), \dots, f_m(x)$  досягають максимуму в одній і тій же точці, то задача (2.1) має ідеальне рішення:

$$x^* = \arg \max_{x \in D} f_i(x), \forall i = \overline{1, m}$$

Однак, наявність «ідеального рішення» а багатокритеріальній задачі – це скоріше виняток з правила, тому при розгляданні задачі (2.1) головною метою визначення принципу оптимальності, чому одне рішення оптимальніше за інші.

Відповідно до цього головним напрямом знаходження рішення для багатокритеріальної задачі (2.1) – це пошук компромісу.

Існуючі методи поділяють на дві головні групи: до першої відносяться методи, які намагаються зменшити кількість критеріїв за допомогою додаткових припущень, друга група намагається зменшити кількість альтернатив для користувача, залишивши лише допустимі альтернативи. До першої групи – методи якої не мають строгої формалізації та визначаються потребами користувача, відносяться методи головного критерію, граничних критеріїв, згортки, відстані.

## 2.1 Метод головного критерію

Метод головного критерію [7] полягає в приведенні багатокритеріальної задачі до однокритеріальної, завдяки вибору головного критерію і введення обмежень на інші. Проте цей метод має можливість бути використаний тільки тоді, коли система має достатньо вихідної інформації, щоб обрати той самий головний критерій. На практиці, важність обраного критерію повинна перевищувати більш ніж у три рази усі інші. Тоді найкраще рішення можливо визначити як:

$$x^* = \arg \max K_0(x), \text{ де } x \in D$$

де  $K_j(x) \geq K_{j\text{порог}}$  для усіх інших критеріїв, тобто їх значення мають бути у дозволених границях.

Обравши головний критерій, для усіх інших критеріїв вводиться система контрольних показників, що зводить рішення до однокритеріальної умовної оптимізації.

Проте, цей метод має ряд глобальних недоліків, що зменшує ймовірність його використання, а саме: надмірне спрощення структури задачі, суб'єктивізм при обиранні головного критерію та обмежень на інші критерії (обирання занадто низьких границь може призвести до Парето не оптимального рішення) та можливість втрати ефекту сукупного впливу декількох другорядних критеріїв.

## 2.2 Методи згортки

Головна мета методів згортки – це згорнути, перетворити множину часткових критеріїв в єдиний супер-критерій, що зводить задачу багатокритеріального вибору до однокритеріальної, а саме до вирішення задачі максимізації цього інтегрального критерію [8].

У загальному вигляді метод згортки приватних критеріїв можна представити таким чином: задається функція згортки  $\varphi(u_1, u_2, \dots, u_m)$  Кількість змінних у функції згортки дорівнює числу приватних критеріїв. Єдиний інтегральний критерій у вихідній задачі формується як суперпозиція функцій приватних критеріїв і функції згортки:

$$F(x) = \varphi(f_1(x), f_2(x), \dots, f_m(x))$$

Отже, задача багатокритеріальної оптимізації зводиться до вирішення задачі максимізації або мінімізації супер-критерію  $F(x)$  на множині  $D$ , отриманого завдяки згортці приватних критеріїв.

Головний недолік методів згортки – це суб'єктивний вибір вагових коефіцієнтів для кожного критерію, адже саме його найважче вирахувати, навіть базуючись на пріоритетах користувача.

При використанні згорток доводиться вирішувати такі проблеми:

- різні діапазони зміни критеріїв;
- різне якісне напрямок критеріїв;
- різні рівні значущості (важливості) для людини, яка приймає рішення, різних критеріїв;
- лінгвістичні (словесні) значення деяких приватних критеріїв.

Найпоширенішими методами згортки є саме адитивна та мультиплікативна.

### 2.2.1 Метод адитивної згортки

Адитивна згортка представляє собою такий собі супер-критерій, що є сумою абсолютних нормованих значень усіх обраних приватних критеріїв у пропорції до їх вагового коефіцієнтів. У цьому випадку супер-критерій зазвичай строїться як зважена сума приватних критеріїв:

$$F(x) = \sum_{i=1}^n a_i f_i(x),$$

де  $a_i$  – це ваговий коефіцієнт, які обираються так, щоб  $\sum_{i=1}^n a_i = 1$ ,  $j$  – кількість критеріїв.

Таким чином отримуємо однокритеріальну задачу математичного програмування – найкращим рішенням є те, що відповідає максимуму загального супер-критерію на множині альтернатив  $D$ .

$$F x^* = \sum_{i=1}^n a_i f_i(x) \rightarrow \max (\min), \text{ де } x \in D, \quad (2.2)$$

Вкрай рідко зустрічаються багатокритеріальні системи з критеріями однієї величини, що вимагає їх нормалізації до абсолютних величин при отриманні узагальненого критерію. Нормований критерій являє собою відношення "натурального" приватного критерію до деякої нормуючої величиною. При цьому вибір нормуючий подільник повинен бути обґрунтований. Можливо кілька підходів до вибору нормування критерію:

- максимальні значення критеріїв, що досягаються в області існування проектних рішень (область D);
- кращі світові досягнення в даній області;
- різницю між максимальним і мінімальним значеннями критерію в області D;
- розраховують завдяки:

$$f_i(x) = \frac{F_i^{max} - F_i(x)}{F_i^{max} - F_i^{min}} \text{ чи } f_i(x) = \frac{F_i(x) - F_i^{min}}{F_i^{max} - F_i^{min}}.$$

### 2.2.2 Метод мультиплікативної згортки

Мультиплікативна згортка базується на принципі справедливої сатисфакції відносних змін приватних критеріїв [9]. Для цього використовується аналогічний підхід як з адитивною згорткою, окрім цільової функції супер-критерію, який має вигляд:

$$F(x) = \prod_{i=1}^n f_i^{a_i}(x),$$

де супер-критерій – це добуток критеріїв  $f_i$ , кожен з котрих зведено у ступінь  $a_i$  – ваговий коефіцієнт, які обираються так, щоб  $\sum_{j=1}^n a_j = 1$ ,  $j$  – кількість критеріїв.

При використанні мультиплікативних критеріїв не потрібно нормувати приватні критерії, і це є їх перевагою. Головною різницею при виборі мультиплікативної або адитивної згортки є важливість зміни абсолютних або відносних значень критеріїв.

### 2.3 Лексикографічна оптимізація

Метод забезпечує лексикографічне впорядкування альтернатив за значеннями критеріїв; при цьому найперша альтернатива є найбільш кращою, друга - менш кращою і так далі. Лексикографічний метод нагадує пошук слова у словнику: спочатку ми обробляємо першу букву, далі наступну і так далі, а саме у відсортованій множині критеріїв від найважливішого, спочатку вирішується задача максимізації критерію по черзі [10]. Спочатку вирішується задача максимізації  $f_1(x)$  на множині  $D$ . Визначимо максимальне значення та  $D_1$  – множину оптимальних рішень цієї задачі:

$$\bar{\mu}_1 = \max_{x \in D} f_1(x), \quad (2.3)$$

$$D_1 = \{ x \in D : f_1(x) \geq \bar{\mu}_1 \},$$

де множина усіх  $x \in D$ , на котрих досягається максимальне значення першого критерію.

На наступному кроці вирішується завдання максимізації наступного за важливістю другого критерію  $f_2(x)$  на множині  $D_1$  і так далі до завдання максимізації останнього критерію  $f_m(x)$ . Таким чином, кожна наступна задача максимізації чергового приватного критерію вирішується на безлічі оптимальних рішень попередньої задачі.

Головним недоліком цього методу є поступова обробка кожного з критеріїв, адже у випадку, коли на перших кроках вирішення задачі було виявлене одне рішення, то наступні критерії, котрі будуть оброблятися далі, не матимуть сенсу і не впливатимуть на результат. Ця проблема призвела до появи модифікованої версії методу – послідовник поступок.

## 2.4 Метод послідовних поступок

Як було зазначено у розділі 2.3, метод послідовних поступок – це розширена версія методу лексикографічної оптимізації, що здійснює невеличкі поступки при обробці кожного з критеріїв. Отже, як і у попередньому методі, спочатку потребується відсортувати критерії за важливістю, так щоб перший був головним, а наступні менш значимі.

Спочатку вирішується задача максимізації найважливішого критерію  $f_1(x)$  на множині  $D$ . Визначимо максимальне значення через вираз (2.3). На наступному кроці робиться поступка по відношенню до оптимального значення першого критерію на деяку величину  $\varepsilon_1 > 0$ , тобто задача максимізації критерію  $f_2(x)$  вирішується на множині  $D_1$  в яке входять не тільки всі оптимальні рішення в задачі максимізації першого критерію, а й ті рішення, у яких значення першого критерію відрізняється від оптимального значення не більше ніж на величину поступки:

$$D_1 = \{ x \in D: f_1(x) \geq \bar{\mu}_1 - \varepsilon_1 \}.$$

Далі на наступному кроці вирішується завдання максимізації другого за важливістю критерію  $f_2(x)$  на множині  $D_1$  і так далі. В якості вирішення вихідної

багатокритеріальної задачі береться рішення останнього завдання максимізації критерію  $f_n(x)$ .

Зважаючи на те, що порядок обробки критеріїв має великий вплив на результат методу та вихідну множину альтернатив, то доцільно використовувати цей метод в задачах, де критерії впорядковані природним чином. Слід зазначити, що суттєвим недоліком є те, що знайдене рішення може бути неоптимальним по Парето.

## 2.5 Побудова множини Парето

Множина Парето [11] – це множина допустимих альтернатив у задачі багатокритеріальної (векторної) оптимізації, для котрих не існує іншої допустимої альтернативи, яка має по усім критеріям не гірше оцінки і хоча б по одному з критеріїв – краще. Оптимальність по Парето говорить про те, що неможливо покращити значення одного критерію, не погіршуючи при цьому значення хоча би одного з інших.

Передбачається, що кожен із критеріїв характеризує якісно відмінний від інших аспект, властивість об'єкта та інше. Так як порівняння різноякісних речей не має сенсу, то упорядкування підлягають тільки ті пари об'єктів, в яких один не гірше іншого за всіма параметрами. Якщо при цьому по одному або декільком критеріям один об'єкт буде кращий за інший, то кажуть, що він домінує. У множині Парето жоден об'єкт не домінує над іншим. Власне, процедура знаходження множини Парето і полягає в знаходженні домінуючих об'єктів і їх виключення з розгляду, що дозволяє знайти компроміс.

Слід відмітити, що цей метод, як і лексикографічний та метод послідовних поступок, відноситься до групи методів, який зменшує кількість альтернатив, а не критеріїв.

## 2.6 Метод «ідеальної» точки

Вирішивши задачу максимізації кожного з критеріїв –  $f_i(x)$ , де  $i = \overline{1, m}$  на множині  $D$ , отримаємо максимальні значення для кожного з приватних критеріїв:

$$w_i^* = \max_{x \in D} f_i(x), i = \overline{1, m}.$$

Таким чином,  $w_i^*$  є максимально можливим значенням  $i$ -го критерію  $f_i(x)$  на множині  $D$ . Точка  $x_i^* \in D$ , така, що  $w_i = f_i(x_i^*) = w_i^*$ , є рішенням однокритеріальної задачі оптимізації.

Допустимо, що  $w^* = (w_1^*, w_2^*, \dots, w_m^*)$ . Точка  $w^*$  називається ідеальною, так як в ній усі критерії мають максимально можливе значення, і отримати більше значення по одному з критеріїв, не зменшуючи значення інших критеріїв – неможливо [12]. Далі вирішується задача пошуку такого вектору  $w$ , для котрого відстань до ідеальної точки  $p(w^*, w)$  мінімальне. В якості функції відстані можливо брати різні метрики, зокрема, середньоквадратичне відхилення  $p(w^*, w) = \sqrt{\sum_{i=1}^m (w_i^* - w_i)^2}$ .

Таким чином, знаходиться найбільш близька до ідеальної точки точка на множині  $W = \{(f_1(x), f_2(x), \dots, f_m(x))\}, x \in D$ . Вона і вважається рішенням задачі багатокритеріальної оптимізації.

## 2.7 Метод справедливого компромісу

Рішення, при котрому зниження відносної якості рішення за однією безліччю критеріїв не перевищує покращення відносної якості рішення за будь-якою іншою

безліччю, називається справедливим компромісом. Слід зауважити, що критерії повинні нормалізовані:

$$F(x) = \max_{x \in D} \prod_{i=1}^n f_i(x_i)$$

Головний недолік цього методу – можливість використовувати лише тоді, коли усі критерії мають однакову вагу.

## 2.8 Результати аналізу методів

Базуючись на інформації отриманої завдяки аналізу основних методів вирішення багатокритеріальних задач, можливо сказати, що для вирішення проблеми, яка полягає у підборі максимально влучної туристичної мандрівки для користувача завдяки визначенню пріоритетності критеріїв та їх ваги, потрібно використовувати методи, які не зменшують кількість альтернатив, цим самим обмежуючи можливість вибору людині, а виконують їх ранжування. Отже, методи лексикографічної оптимізації, послідовних поступок, «ідеальної» точки, побудови множини Парето – не мають практичного використання у туристичній предметній галузі.

Розглядаючи іншу групу методів, маємо можливість відзначити, що велика кількість критеріїв не дозволяє обрати границі допустимого використання кожного з критеріїв, що відкидає використання таких методів, як метод порогових критеріїв та метод головного критерію, який використовує порогові границі для другорядних критеріїв. Метод справедливого компромісу не може бути використаним, адже усі

критерії повинні мати однакову вагу, що суперечить головній меті даної роботи, дозволяти користувачам регулювати важливість критерію за їх потребою.

Виходячи з цього, найвлучнішими з проаналізованих методів є методи згортки – адитивна та мультиплікативна. А отже, на підставі інформації отриманої по цим методам, можна відмітити, що мультиплікативна згортка є менше влучною до туристичної предметної галузі, адже якщо вага одного з критеріїв буде прагнути до нуля, то і коефіцієнт ранжування буде прагнути до 0, не зважаючи на інші критерії та їх пріоритети. З цього можемо зробити висновок, що якщо користувач відмітить критерій, як неважливий і дасть йому 1% ( $a_i = 0.01$ ) ваги, то навіть найкращий варіант, де буде зібрано усі побажання користувача по заданим критеріям, потягне підсумковий коефіцієнт ранжування в низ таблиці, що не є найкращим варіантом.

Отже, найкращим методом для виконання ранжування у додатку було обрано адитивну згортку. Вона вимагає нормування критеріїв, які в даній системі не мають однієї величини, що призводить до безмірної середи вимірювання, та розробку алгоритму для підрахування відносної ваги для кожного критерія, що має відносну вагу, задану користувачам та розташований за пріоритетністю.

### 3 ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

Розвиток сучасних інформаційних технологій призводить до постійного збільшення складності інформаційних систем, що призводить до ускладнення розуміння усіх процесів, які виконує сам продукт, навіть якщо їх декілька. Для прозорішого розуміння програмного продукту виконується моделювання цієї системи.

Моделювання програмного продукту призводить до кращого розуміння програмної системи та усіх її процесів. Кожен етап моделювання, описує один з основних аспектів системи, що понижує складність розуміння додатку для людини, завдяки наочній графічній формі. Поєднуючи усі етапи моделювання буде отримана система, що описана з різних сторін та відображає різні аспекти, а отже і усю систему загалом.

Для демонстрації дослідження проведеного у кваліфікаційної роботи треба спроектувати систему, де визначити наступні етапи:

- етап UML моделювання, що відображають усі процеси системи та взаємодію користувача системи з додатком;
- етап проектування архітектури додатку, що відображає усі внутрішні процеси додатку, а саме взаємодію між усіма частинами системи;
- етап проектування алгоритму ранжування на основі адитивного методу згортки, що включає в себе визначення алгоритму для нормалізації критеріїв, спосіб визначення вагових коефіцієнтів для кожного з критеріїв та саме розробка алгоритму.

На етапі проектування додатку немає етапу проектування бази даних, так як головна мета дослідження – це оптимізації пошуку туристичних мандрівок в умовах карантину, а отже демонстраційний додаток буде отримувати список мандрівок зі

стороннього сервісу та проводити їх ранжування, після чого віддавати клієнту, тобто не буде ніякої збереженої інформації, а отже і бази даних.

### 3.1 UML-проекування програмної системи

Для наглядного відображення структури та функціоналу додатку, що буде демонструвати ранжування туристичних подорожей за обраними критеріями та їх пріоритетністю, розробленого під час виконання кваліфікаційної роботи, буде використана мова UML (уніфікована мова моделювання) [13].

Головною метою дослідження є оптимізація пошуку туристичних мандрівок під час карантину, а саме їх ранжування за поданими критеріями, а отже додаток повинен дозволяти користувачеві обрати набір критеріїв для мандрівки та отримати підходящі подорожі, які розташовані у порядку підрахованому алгоритмом ранжування. Загальну інформацію про функціональність демонстраційного додатку відображено в діаграмі варіантів використання (рис. 3.1).

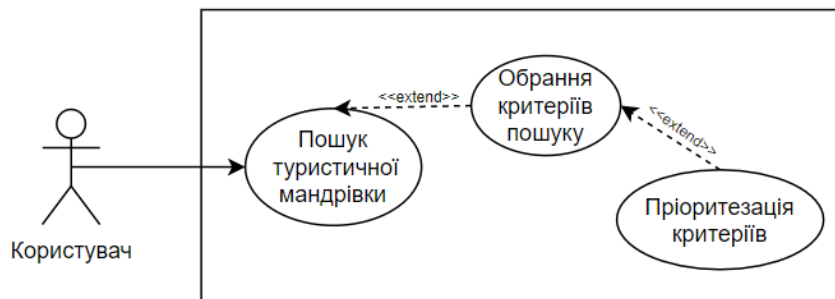


Рисунок 3.1 – Діаграма варіантів використання

Актором на цій діаграмі є користувач, який має можливість обрати критерії пошуку та розмістити їх у черзі пріоритетності, що дозволить системі розташувати відповідні альтернативи в порядку ранжування системою.

Логіку виконання ранжування подорожей відобразимо на діаграмі послідовності (рис. 3.2), що дозволить простежити усі кроки, які виконує система для отримання туристичних мандрівок у відповідній черзі. На діаграмі можливо побачити, що користувач вводить вхідну інформацію про критерії пошуку, після чого клієнтський додаток надає дані серверній частині. Сервер, що виконує основну логіку ранжування, отримує відповідні дані з бази даних, котрі опрацьовує, використовуючи алгоритм ранжування, та відає клієнтській частині, що і відображає результат для користувача, де зверху розташовані варіанти максимально підходящі потребам користувача.

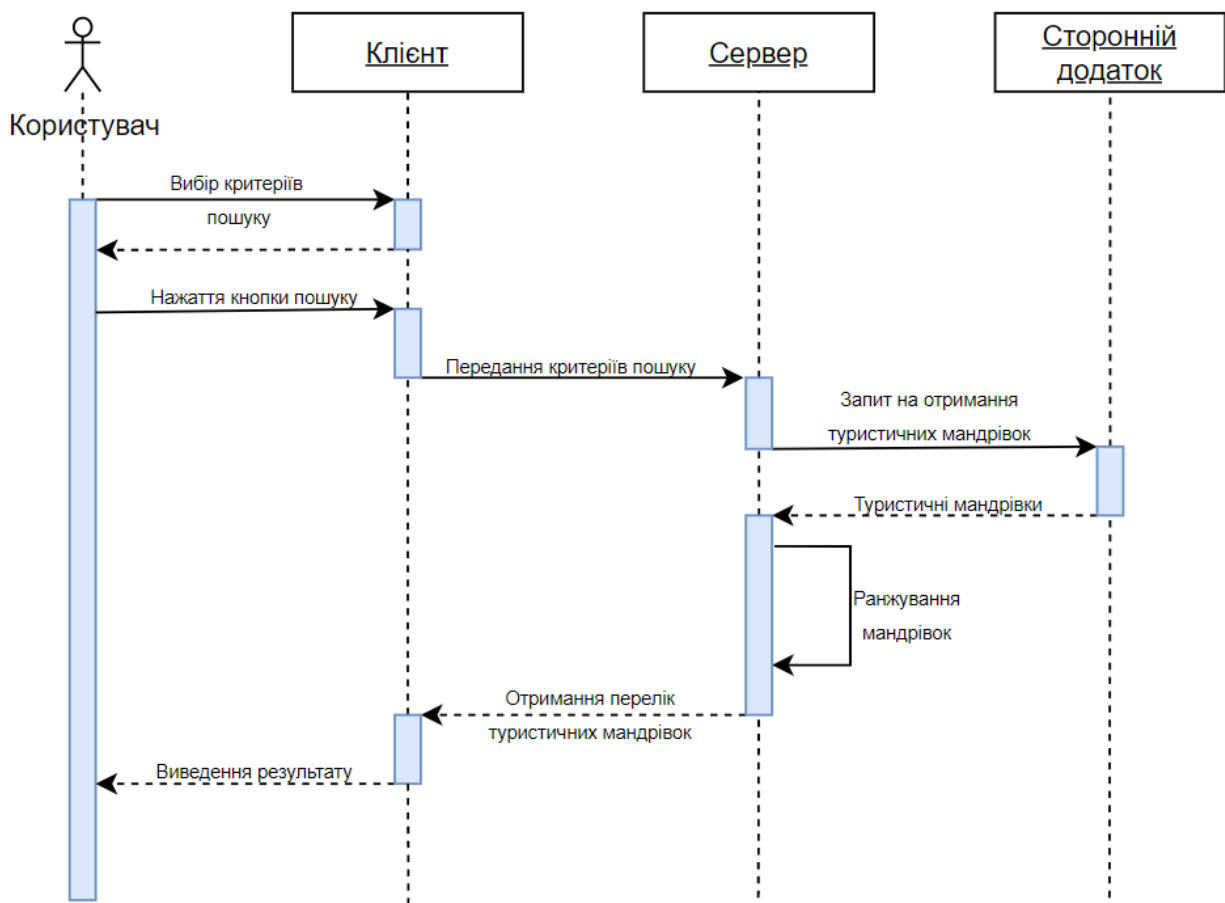


Рисунок 3.2 – Діаграма послідовності

На підставі отриманих даних про додаток, можливо зробити висновок, що для демонстрації алгоритму ранжування потрібно розробити одно-сторінковий додаток,

де буде два головних компоненти – фільтр-панель, для надання можливості обрати критерії пошуку та сортування їх пріоритетності, та область для відображення відповідних туристичних мандрівок у порядку, відповідно алгоритму ранжування.

### 3.2 Проектування архітектури програмної системи

Для реалізації додатку було вирішено використовувати клієнт-серверну архітектуру, яка є найбільш надійною у подібного роду системах, а саме у розподілених системах мереживних додатків, які потребують здійснення обміном даних. Архітектуру системи та взаємодію компонентів наведено на рисунку 3.3.

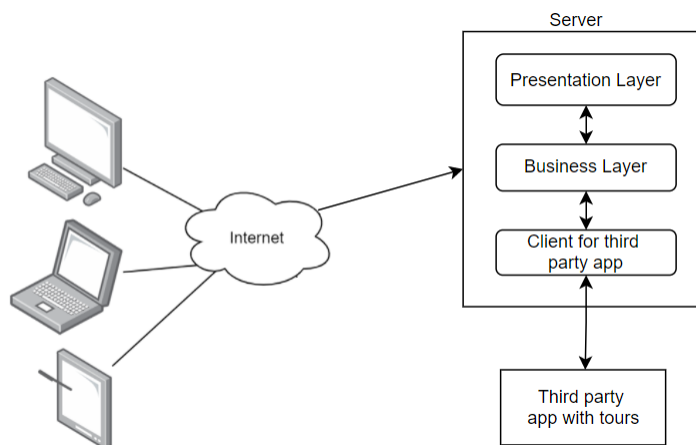


Рисунок 3.3 – Архітектура системи

З назви витікає дві головні сутності – клієнт та сервер, а саме їх множина та мережа, яка забезпечує взаємодію між ними. Така система дозволяє убезпечити систему від виходу зі строю, адже коли один сервер не має змоги відповісти запиту перенаправляється на працездатний сервер і користувач не помічає проблеми, а технічні спеціалісти мають час на її виправлення. Отже, сервери працюють

паралельно, як і клієнти. Слід відмітити, що за даною схемою клієнт повинен починати зв'язок, а сервер лише відповідати на запити клієнта.

Для розробки серверної частини демонстраційного додатку була обрана класична багатошарова архітектура, кожен шар якої приховує певну частину логіки додатку, а отже робить його більш гнучким. В проекті було виділено два головних шари: шар бізнес логіки – рівень, який приховує у собі головну частину дослідження, а саме обробку та ранжування інформації отриманої з стороннього додатку та шар представлення, а саме API додатку, куди будуть надходити запити від клієнтів.

Клієнтський додаток, який контролюється користувачем за допомогою браузера, відправляє запит до API додатку, який направляє запит далі до потрібної логіки та після обробки повертає результат до клієнта, що відображає його на привабливому та зрозумілому користувачеві інтерфейсі.

### 3.3 Розробка алгоритму ранжування альтернатив

Головною метою кваліфікаційної роботи є оптимізація пошуку [15] туристичних мандрівок, а саме ранжування альтернатив, що допоможе користувачеві зробити вибір швидше базуючись на його пріоритети.

Завдяки аналізу, проведеного у розділі 2, було вирішено використовувати алгоритм адитивної згортки, який не зменшує кількість альтернатив та не надає одному з критеріїв повної влади над результатом, що робить результуючу упорядкованість більш підходящою для користувачів.

На підставі виразу (2.2), де відображено математичну модель для адитивної згортки, можна відмітити два головних елементи – ваговий коефіцієнт ( $a_i$ ) та значення критерію ( $f_i(x)$ ), а саме нормалізованого.

А отже, сам алгоритм складається з трьох кроків: нормалізація критеріїв, визначення вагових коефіцієнтів за пріоритетністю та саме згортка критеріїв до супер-критерію. Від користувача метод адитивної згортки потребує лише визначення пріоритетності, а саме ваги кожного з критеріїв, що допоможе персоналізувати результат саме для цієї людини за її потребами.

### 3.3.1 Нормалізація критеріїв

Даний крок є важливий для забезпечення коректної формалізації процесу прийняття рішень [16], адже частіше за все критерії, які використовуються у задачах багатокритеріального прийняття рішень, мають різні одиниці вимірювання та діапазони змін. Одні критерії можуть задаватися за кількісною шкалою та мати розкид значень в тисячі гривень, як вартість мандрівки; другі – визначати в якісній шкалі вимірювання; інші – мати різницю у долях відсотку та так далі. Однак, більшість принципів оптимальності, одним з яких є і адитивна згортка, мають сенс лише тоді, коли критерії мають однакові одиниці вимірювання масштабу. Отже, це призводить до того, що необхідно перетворити усі критерії до єдиної шкали, одиниці вимірювання та рівним розмахом варіацій значень – нормалізації критеріїв. Частіше за все використовується безмірна шкала з діапазоном від 0 до 1.

Існує велика безліч методів нормалізації відповідно до умов поставленої задачі, але усі вони мають поділ на максимізацію ( $f'_{max}$ ) та мінімізацію ( $f'_{min}$ ) значень критеріїв, адже у певних випадках для рішення задачі важливо отримати максимальне значення критерію і чим більше тим краще, а в інших – навпаки. У результаті чого ми отримуємо безмірне значення для кожного критерію, що і використовується адитивною згорткою у перетворенні часткових критеріїв до супер-критерію.

Розглянемо декілька найпоширеніших методів нормалізації [17]:

– порівняльна або відносна нормалізація – нормалізація, яка використовується, коли відомі ідеальні (еталонні) значення критерію. Однак, частіше за все замість цих значень використовується максимальне або мінімальне значення критерію у вибірці:

$$f'_{max} = \frac{f_j(x)}{\max f(x)},$$

$$f'_{min} = \frac{\min f(x)}{f_j(x)}$$

де  $f_j(x)$  – значення критерію для альтернативи,  $\max f(x)$  – максимальне значення критерію у вибірці;

– природна нормалізація – нормалізація, яка представляє собою комбінацію методів абсолютної та відносної нормалізації, відображає перетворення елемента на основі порівняння його значення з мінімальним значенням параметру і оцінки інтервалу зміни:

$$f'_{max} = \frac{f_j(x) - \min f(x)}{\max f(x) - \min f(x)} \quad (3.1)$$

$$f'_{min} = \frac{\max f(x) - f_j(x)}{\max f(x) - \min f(x)} \quad (3.2)$$

де  $f_j(x)$  – значення критерію для альтернативи,  $\max f(x)$  – максимальне значення критерію у вибірці,  $\min f(x)$  – мінімальне значення критерію вибірки;

– нормалізація усереднення – демонструє відносний вклад критерію в підсумковий показник, адже це відношення значення критерію до суми усіх критеріїв:

$$f'_{max} = \frac{f_j(x)}{\sum_j f_j(x)}, \quad (3.3)$$

$$f'_{min} = \frac{1/f_j(x)}{\sum_j (1/f_j(x))},$$

де  $f_j(x)$  – значення критерію для альтернативи,  $\sum_j f_j(x)$  – сума значень усіх критеріїв;

– векторна нормалізація – це відношення значення критерію до значення відстані вектору, що отриманий як сума усіх значень (векторів):

$$f'_{max} = \frac{f_j(x)}{\sqrt{\sum_j f_j^2(x)}},$$

$$f'_{min} = 1 - \frac{f_j(x)}{\sqrt{\sum_j f_j^2(x)}}$$

де  $f_j(x)$  – значення критерію для альтернативи,  $\sum_j f_j^2(x)$  – сума квадратів значень усіх критеріїв;

Кожен з цих видів нормалізації може бути використано для зведення критеріїв до єдиної шкали і не має особливих відмінностей, окрім по відношенню до чого буде виконуватись нормалізація. Отже, для кваліфікаційної роботи будемо використовувати природну нормалізацію, що визначена у виразах (3.1) та (3.2), адже ця нормалізація дозволяє витримувати діапазон значень від 0 до 1 та рівномірно розподіляє значення критеріїв, адже базується на існуючих максимальних та мінімальних значеннях у вибірці, що охоплює весь діапазон значень підходящих альтернатив.

### 3.3.2 Визначення вагових коефіцієнтів

Кожен вибір здійснюється на підставі набору даних, де щось є першочерговим, а якісь з критеріїв мають невагомні коефіцієнти. Саме ці дані і допомагають людині обрати максимально влучний варіант для неї, зіставивши усі вхідні параметри, а отже уся ця інформація потрібна і системі пошуку туристичних мандрівок для їх ранжування. Відповідно до цього, кожен обраний критерій повинен мати свою вагу у цій системі, що відображає його значимість на вплив на результат згортки.

Значення вагових коефіцієнтів кожного з критеріїв обирається так, що:

$$0 < p_i < 1, i = 1 \dots k, p_1 + p_2 + \dots + p_k = 1,$$

де  $p_i$  – ваговий коефіцієнт критерію.

Отже, потребується нормалізувати пріоритети, які задає користувач візуально за допомогою пересування критеріїв між собою, до вагових коефіцієнтів. Для цього будемо використовувати нормалізацію усереднення, яка виражена у виразі (3.3), адже саме вона відображає вклад критерію у підсумковий показник, що і відображає ваговий коефіцієнт.

Для нормалізації усереднення спочатку потрібно перетворити пріоритет на значення ваги – чим нижче пріоритет, тим нижче значення [18]. Тобто, ми маємо  $k$  критеріїв і критерію з першим пріоритетом задаємо значення  $k$ , наступному пріоритету значення –  $k - 1$ , і так до  $k$ -пріоритету де буде значення 1. Далі використовуємо нормалізацію усереднення, тобто ділимо значення пріоритету на суму усіх отриманих значень.

Отже, визначивши спосіб нормалізації критеріїв та нормалізації і перетворення пріоритетів, маємо можливість переходити до наступного кроку – згортки.

### 3.3.3 Використання адитивної згортки

Добре розроблений математичний апарат для вирішення завдань оптимізації з одним критерієм робить привабливою ідею заміни множини критеріїв (векторів) деяким узагальненим скалярним критерієм, який би повністю відображав в собі усі якості рішення, закладені в окремих критеріях. Одним з методів, виконуючих преображення множини критеріїв до супер-критерію під час пошуку компромісу, і є адитивна згортка, яка була обрана як метод розв'язання багатокритеріальної задачі вибору туристичної мандрівки у розділі 2.

Для використання адитивної згортки потребуються нормалізовані значення критеріїв та вагові коефіцієнти критеріїв отриманих зі значень пріоритету, методи для яких було визначено у розділах 3.3.1 та 3.3.2 відповідно.

Розглянемо покрокове вирішення задачі вибору туристичної мандрівки адитивною згорткою за допомогою ранжування. Допустимо, що маємо наступний набір альтернатив у таблиці 3.1.

Таблиця 3.1 – Початкова множина альтернатив

Альтернатива/ Критерій	Вартість, (грн)	Кількість зірок готелю, (одиниці)	Корт для тенісу, (+ або -)	WI-FI в номері, (+ або -)
A1	900	2	+	-
A2	800	3	-	-
A3	1300	5	-	+
A4	1400	4	+	-
A5	1000	3	-	+
A6	1200	4	-	+

Більшість критеріїв мають різну систему вимірювання: гривні, одиниці, існує або не існує, тощо, – це демонструє явну потребу в нормалізації значень критеріїв до абсолютних, безмірних.

Відповідно до кожного з критерію, визначимо максимізацію чи мінімізацію прагнемо досягти. В даному прикладі усі критерії, окрім вартості потребують максимізації, адже ми хочемо мати, як змога більше зірок, мати корт для тенісу та WI-FI в номері. Але все це за мінімальну вартість, що визначає, що критерій вартості потребує мінімізації при нормуванні. Отже, критерії: кількість зірок, наявність корту для тенісу та WI-FI в номері, – будуть нормалізуватись за формулою (3.1), а вартість – за формулою (3.2).

Результат нормування наведено у таблиці 3.2.

Таблиця 3.2 – Нормалізовані значення критеріїв

Альтернатива/ Критерій	Вартість	Кількість зірок готелю	Корт для тенісу	WI-FI номері	в	Результат згортки без вагових коефіцієнтів	Номер ранжування
A1	0.83(3)	0	1	0		1.83(3)	4
A2	1	0.3(3)	0	0		1.3(3)	6
A3	0.16(6)	1	0	1		2.16(6)	1
A4	0	0.6(6)	1	0		1.6(6)	5
A5	0.6(6)	0.3(3)	0	1		1.9(9)	2
A6	0.3(3)	0.6(6)	0	1		1.9(9)	3

Немаючи вагових коефіцієнтів, маємо можливість скласти отриманні нормовані значення, адже вони вже безмірні і відсортувати за отриманим скалярним значенням. У цьому випадку альтернатива A3 є найпривабливішою для користувача, а A2 найменш підходящою.

Наступним кроком є визначення вагового коефіцієнту критерію за пріоритетом обраним користувачем. Для кожної людини, той чи інший критерій впливає по різному, а отже має різний пріоритет для людини або вагу для згортки. Припустимо, що маємо наступну пріоритетність задану користувачем: 1-ий пріоритет – «корт для тенісу», 2-ий – «вартість», 3-ій – «кількість зірок готелю», 4-ий – «WI-FI в номері». Використовуючи нормалізацію усереднення у випадку максимізації, що виражена у формулі (3.3), адже більший пріоритет критерію має найвпливовіше значення, та

перетворення номер пріоритету у значення його вагомості, отримуємо наступну таблицю 3.3.

Таблиця 3.3 – Нормалізація пріоритетності критеріїв

Критерій	Пріоритет	Значення пріоритету	Результат нормалізації
Вартість	2	3	0.3
Кількість зірок готелю	3	2	0.2
Корт для тенісу	1	4	0.4
WI-FI в номері	4	1	0.1

Отже, отримавши нормалізовані значення критеріїв та їх вагові коефіцієнти (див. таблицю 3.4) маємо змогу виконати згортку для кожної з альтернатив за формулою (2.2).

Таблиця 3.4 – Результат адитивної згортки з ваговими коефіцієнтами

Альтернатива/ Критерій	Вартість, (0.3)	Кількість зірок готелю, (0.2)	Корт для тенісу, (0.4)	WI-FI в номері, (0.1)	Результат згортки з ваговими коефіцієнтами	Номер ранжування
A1	0.25	0	0.4	0	0.65	1
A2	0.3	0.06(6)	0	0	0.36(6)	3
A3	0.05	0.2	0	0.1	0.35	5
A4	0	0.13(3)	0.4	0	0.53(3)	2
A5	0.2	0.06(6)	0	0.1	0.36(6)	4
A6	0.1	0.13(3)	0	0.1	0.33(3)	6

У результаті виконання згортки та сортуванні значень, отримуємо, що найпривабливішою є альтернатива A1 та найменш привабливою A6, що повністю відрізняється від ранжування нормованих критеріїв без пріоритету, адже враховують усі побажання користувача в пошуку компромісу.

## 4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

### 4.1 Опис засобів розробки

На етапі проектування системи було вирішено використовувати клієнт-серверну архітектуру, що передбачає існування клієнта и сервера, окремими додатками, які незалежні один від одного, окрім відкритого API, яке використовує клієнт.

Відповідно до цього, для серверної частини було використано платформу .Net та мову програмування C#, фреймворк WEB API [19], який використовується для розробки HTTP-сервісів. Дана платформа має можливість працювати з високонавантаженими додатками, а саме це і відбувається з туристичними сайтами.

Для клієнтської частини було обрано сучасний фреймворк Angular 9 з використанням мови програмування Typescript для реалізації логіки додатку. Даний фреймворк був обраний завдяки своїй компонентній структурі, яка дозволяє уникати дублювання коду за допомогою створювання компонентів та їх перевикористанню в різних місцях. Також даний фреймворк використовує SPA (Single Page Application) [20], а тобто односторінковий веб-додаток, який динамічно підгружає потрібні частини сторінки за допомогою JavaScript. На практиці це визначається, що при переході або прокрутці сайту, замість повного перезавантаження сторінки відбувається заміна або завантаження інших компонентів системи, що покращую, а саме пришвидшує, працездатність системи. Для розмітки сторінки використано HTML та CSS для надання привабливого вигляду сторінкам [21], фреймворк Twitter Bootstrap для привабливого та адаптивного відображення інтерфейсу на будь-яких гаджетах та бібліотека jQuery, яка полегшує взаємодію з DOM об'єктами на сторінці.

Отже, клієнтська частина розроблена так, щоб відповідати усім потребам користувача та бути інтуїтивно зрозумілою.

## 4.2 Опис програмної системи

Відкриваючи додаток, користувач бачить головно сторінку сайту, яка інтуїтивно поділена на три частини, а саме шапку (header додатку), основну частину та нижній колонтитул (footer додатку), де відображено контактну інформацію.

Шапка сайту (див. рис. 4.1) надає людині можливість зробити швидкий пошук у систему, а саме обрати країну та за необхідністю її міста або усі існуючі, дату подорожі та кількість людей – дорослих та дітей, що будуть приймати участь у подорожі. Після того, як кнопка «Search» буде нажата, сайт автоматично опустить користувача до частину з переглядом туристичних мандрівок за обраним напрямом та фільтр-панеллю.



Рисунок 4.1 – Шапка додатку

Частина сайту, де відображуються туристичні мандрівки є головною та умовно поділяється на дві частини – фільтр-панель та місце для відображення підходящих варіантів (див. рис. 4.2). Частина з відображенням переліком відповідних варіантів інтуїтивно зрозуміла, адже варіанти розташовані в черзі за умов відповідності критеріям та вподобанням користувача в їх пріоритетності. Частина, що відображає фільтр-панель має більш складний вигляд, адже вміщують в собі перелік фільтрів, а саме: обрати діапазон вартості, кількість зірок (клас) готелю, дальність до центру міста, додаткові послуги готелю: Wi-Fi, послуги няні, той чи інший вид басейну,

наявність тенісного корту, підігрів басейну, тощо, та кнопка для пошуку мандрівок – «Search». Після того, як що найменше один критерій буде обраний користувачем, рядом з кнопкою пошуку появиться кнопка для задання пріоритетності для обраних критеріїв та відобразиться підказка про таку можливість. Якщо користувач не має часу на це, то він має можливість пропустити це і тоді кожен критерій буде вважатися рівнозначним, що допоможе зробити ранжування для відповідних мандрівок. У варіанті підходящих мандрівок, користувач має можливість побачити назву туру, опис, дати прибуття та відправлення, кількість зірок у готелю та ціну туру. Усю іншу інформацію про готель можна переглянути розгорнувши конкретну подорож.

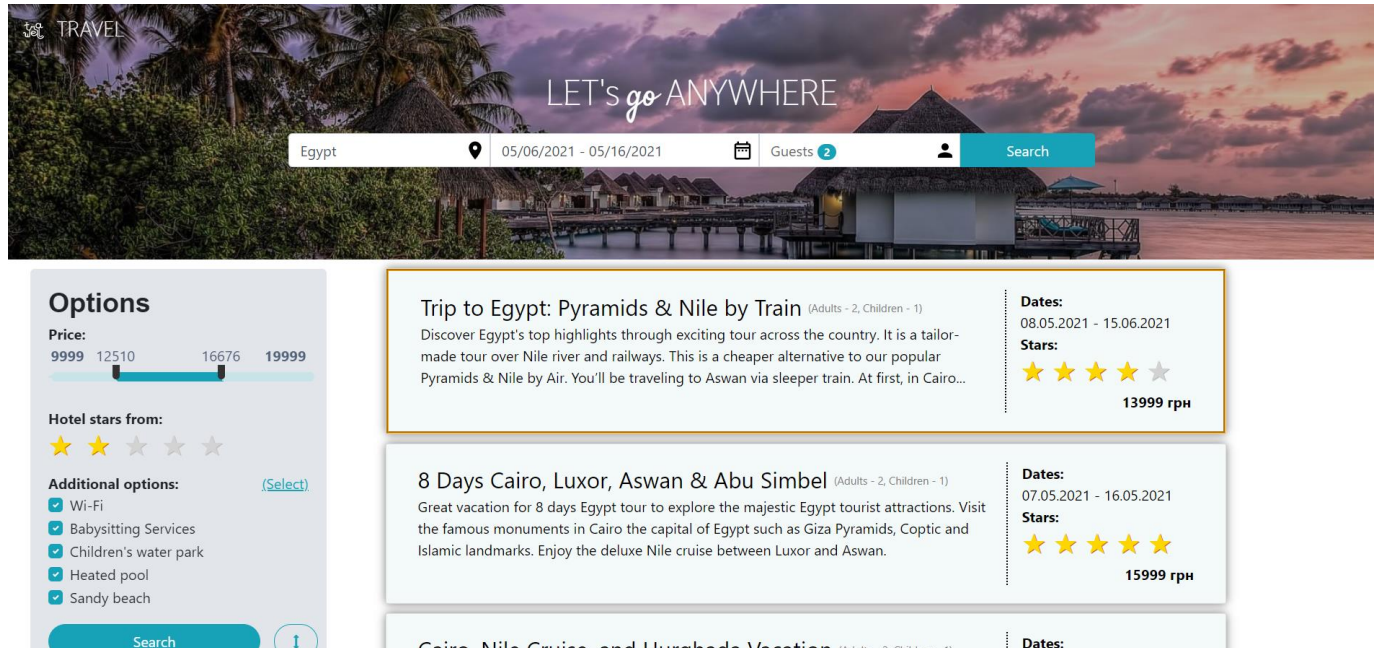


Рисунок 4.2 – Відображення туристичних мандрівок та фільтр-панелі

Якщо ж користувач хоче максимально влучно підібрати тур та зекономити свій час на перегляд усіх можливих варіантів, він може обрати опцію пріоритизації критеріїв.

Відповідно після натискання на кнопку для пріоритизації критеріїв, людина побачить спливаюче вікно з вже обраними критеріями (див. рис. 4.3). Далі завдяки

можливості перетягнути критерії вгору чи вниз, користувач має можливість визначити його пріоритет потреби.

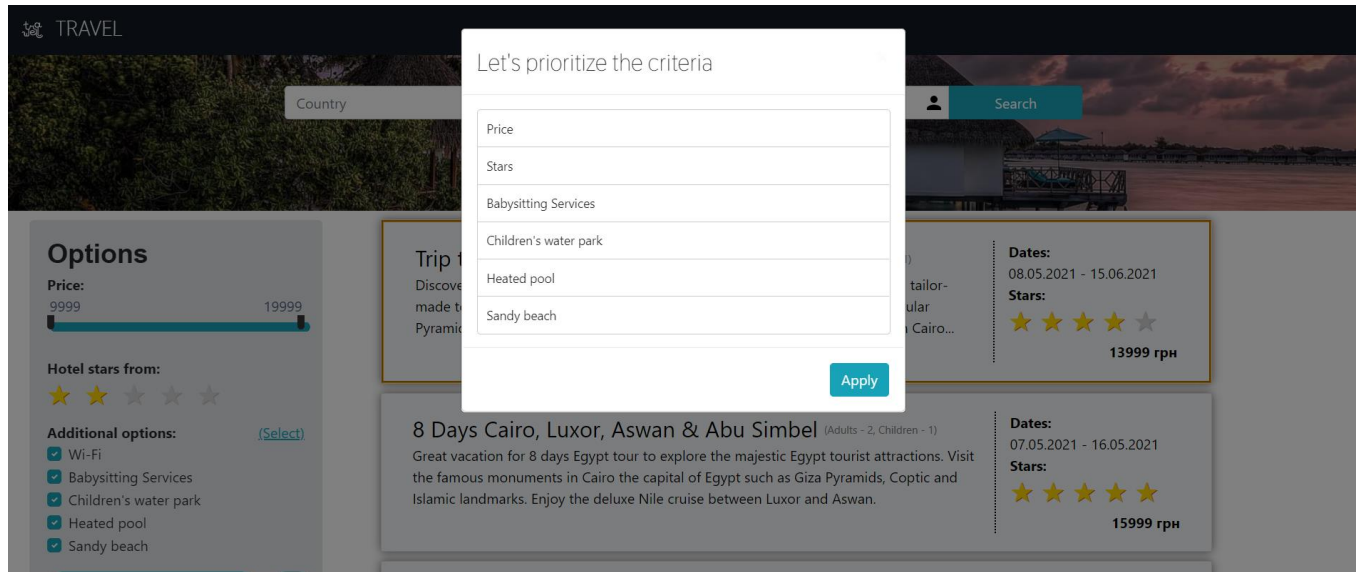


Рисунок 4.3 – Вікно з критеріями для їх пріоритизації

Після натискання кнопки «Apply» для підтвердження обраних пріоритетів, відразу почнеться пошук – користувач побачить відповідні варіанти, відсортовані в порядку вподобанням критеріїв та ті варіанти, що максимально влучні до них будуть відмічені жовто-багряною рамкою, що спростить вибір користувача та вкаже на максимально привабливий варіант.

## ВИСНОВКИ

В рамках написання кваліфікаційної роботи було спроектовано та розроблено веб-систему, яка демонструє роботи алгоритму ранжування даних для пошуку туристичної мандрівки, розробленого під час дослідження вирішення проблеми прийняття рішень у багатокритеріальній середі.

Спочатку було проведено аналіз предметної області, а саме розглянуто декілька відомих додатків для пошуку туристичної мандрівки. На підставі виявленої інформації, а саме переваг та загальних недоліків, було сформовано головну мету дослідження – вирішення проблеми прийняття рішення вибору туристичної мандрівки за заданими критеріями користувача.

Під час дослідження методів вирішення проблем багатокритеріального вибору було розглянуто адитивна, мультиплікативна згортки, множина Парето, метод головного критерію, ідеальної точки, граничних критеріїв та інші, й визначено їх переваги та недоліки для туристичної галузі та користувача. Було визначено дві головні групи методів: перша – зменшує кількість альтернатив для користувача, друга – зменшує кількість критеріїв та проводить ранжування альтернатив. На підставі цих даних, було визначено, що перша група не є підходящою, адже обмежую вибір користувача, урізуючи можливість його вибору. Отже, було вирішено використовувати методи, що зводять усі критерії до супер-критерію та проводять ранжування, а після аналізу цих методів було обрано адитивну згортку.

На основі отриманих результатів, було розроблено алгоритм адитивної згортки, що складається з трьох кроків: нормалізування критеріїв, а саме зведення їх до безмірної шкали; вирахування вагового коефіцієнту на підставі пріоритетності обраної користувачем; та й саме виконання згортки, зведення усіх критеріїв до скалярного значення, яке і використовується у подальшому для ранжування мандрівок.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Система інтернет-бронювання Expedia. URL: <https://www.expedia.com/> (дата звернення: 20.03.2021).
2. Система інтернет-бронювання готелів Booking.com. URL: <https://www.booking.com/> (дата звернення: 20.03.2021).
3. Онлайн-площадка для розміщення та пошук короткострокова оренди Airbnb. URL: <https://www.airbnb.com/> (дата звернення: 20.03.2021).
4. Лотов А.В., Поспелова И.И. Багатокритеріальні задачі прийняття рішень: Уч. збірник. М.: Издат. отдел ф-та ВМиК МГУ, МАКС Пресс, 2008. – 197 с.
5. Richardson L., Amundsen M. RESTful Web APIs: Services for a Changing [Текст]/L. Richardson. – С.: O'Reilly Media, 2013. – 406 с.
6. Штойер Р. Багатокритеріальна оптимізація. Теорія, обчислення, і додатки. – М.: Радіо та зв'язок, 1992.- 504 с
7. Мазалов В.В. Математическая теория игр и ее приложения. – М.: Лань, 2010. – 448 с.
8. Нейман Дж., Моргенштерн О. Теория игр и экономическое поведение / Пер. с англ. — М.: Наука, 1970. — 708 с
9. Грицюк Ю.І., Грицюк М.Ю. Особливості мультиплікативного згортання часткових критеріїв в узагальнений показник. Зб. наук.-техн. пр.: Науковий Вісник НЛТУ України. – 2014. – Вип. 24.11. – С. 341 – 352.
10. Delgado M., Verdegay J. L., Vila M. A. Linguistic decision-making models.//International Journal of Intelligent System. 2012. № 7. P. 479 – 492
11. Ногин В. Д. Множина та принцип Парето. - СПб: Видавничча-поліграфічна асоціація вищих навчальних закладів,2020, 100 с.
12. Oleksandr Topchii, Oleksandr Samantsov, Oksana Mazurova, Mariia Shirokopetleva. A Study of Optimization Models for Creation of Artificial Intelligence for

The Computer Game in The Tower Defense Genre. Problem of Infocommunications. Science and Technology (PIC S&T'2020), Kharkiv, Ukraine.- 6-9 October 2020.

13. Мартін Фаулер. UML. Основи. Короткий посібник з стандартному мови об'єктного моделювання: Пер. с англ. — М.: Наука, 2003. — 208 с.

14. Мартін Фаулер. Архитектура корпоративных программных приложений: Пер. с англ. — Російська редакція, 2004. — 544 с.

15. A. Yerokhin, A. Nechyporenko, O. Turuta, A. Babii. A Lars-Based Method of the Construction of a Fuzzy Regression Model for the Selection of Significant Features Cybernetics and Systems Analysis, vol 52, no 4, pp 641–646, 2016.

16. Соломатін А.Н., Соколова І.І. Методи прийняття рішень: Навчальний посібник. - М.:МНІТ, 2002. - 120 с

17. Економічний ризик: ігрові моделі: Навч. посібник / В. В. Вітлінський, П. І. Верченко, А. В. Сігал, Я. С. Наконечний; За ред. д-ра екон. наук, проф. В. В. Вітлінського. — К.: КНЕУ, 2002. — 446 с.

18. Armstrong R. D., Cook W. D., Seiford L. M. Priority ranking and consensus formation.//Management Science. 2012. № 28. P. 638 – 645

19. Jamie Kurtz. ASP.NET Web API 2: Building a REST Service from Start to Finish. — Paperback, 2014. — 295 с.

20. Michael Mikowski, Josh Powell. Single Page Web Applications. — Manning Publications Co., 2013. — 432 с.

21. Jon Duckett. HTML and CSS: Design and Build Websites. — Paperback, 2011. — 512 с.