

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Центр післядипломної освіти _____
(повна назва)

Кафедра _____ Програмної інженерії _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____ другий (магістерський) _____
(рівень вищої освіти)

Дослідження задачі багатокритеріального вибору на прикладі планування подорожі
(тема)

Виконав: студент 2 курсу, групи ПІЗмзд-17-1 _____
спеціальності 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)

Освітньо-наукової програми
Інженерія програмного забезпечення _____
(повна назва освітньої програми)

_____ Варвашенко Д.А. _____
(прізвище, ініціали)
Керівник _____ доц. Лановий О.Ф. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____

З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет Центр післядипломної освіти

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення

(код і повна назва)

освітньо-наукова програма Інженерія програмного забезпечення

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 20 ____ р.

**ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ**

студентові Варвашенку Дмитру Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження задачі багатокритеріального вибору на прикладі планування подорожі

затверджена наказом по університету від “22” квітня 2019 р № 65 Стз

2. Термін подання студентом роботи до екзаменаційної комісії
05 червня 2019 р.

3. Вихідні дані до роботи методи вирішення задачі багатокритеріального вибору, діаграма архітектури системи. Використовувати ОС Linux, середовище об'єктно-орієнтованого проектування

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, огляд методів вирішення задачі багатокритеріального вибору, аналіз можливих типів архітектури системи, побудування UML діаграм

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Мета завдання, обґрунтування доцільності розроблення, постановка задачі, об'єктна модель системи, базові моделі, методи й алгоритми, UML-діаграми, архітектура системи, _____ демонстраційні матеріали

6 Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка*
1.	Аналіз предметної галузі	19 квітня 2019р.	
2.	Огляд існуючих методів	27 квітня 2019р.	
3.	Методи розв'язання задачі	10 травня 2019р.	
4.	Підготовка пояснювальної записки	25 травня 2019р.	
5.	Спецчастина	26 травня 2019р.	
6.	Підготовка презентації та доповіді	28 травня 2019р.	
7.	Попередній захист	30 травня 2019р.	
8.	Нормоконтроль, рецензування	02 червня 2019р.	
9.	Занесення диплома в електронний архів	03 червня 2019р.	
10.	Допуск до захисту у зав. Кафедри	04 червня 2019р.	

Дата видачі завдання 11 лютого 2019 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Лановий О.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 71 с., 18 рис., 3 табл., 15 джерел, 1 додаток.

АЛЬТЕРНАТИВА, АРХІТЕКТУРА СИСТЕМИ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, КРИТЕРІЙ, ПОДОРОЖ, МАТЕМАТИЧНА МОДЕЛЬ, МЕТОДИ ЗГОРТКИ, REST.

Метою роботи є дослідження методів розв'язання задачі багатокритеріального вибору на прикладі планування подорожі, а також розробка програмної системи для пропонування користувачеві оптимального плану подорожі.

В результаті роботи були проаналізовані методи пошуку кращої альтернативи, та спроектовано систему для пропонування користувачеві оптимального плану подорожі, яка матиме клієнт-серверну архітектуру та спілкуватиметься зі сторонніми системами за допомогою принципу REST.

ALTERNATIVE, CLIENT-SERVER ARCHITECTURE, CONVOLUTION METHODS, CRITERION, MATHEMATICAL MODEL, SYSTEM ARCHITECTURE, REST, TRAVEL.

The aim of the work is to study the methods to solve multicriteria choice task for trip planning and also to design a system to propose the most optimal solution for trip planning.

As a result of the work, the methods of finding the best alternative were analyzed and there was designed a system for trip plan proposing that will have client-server architecture and will interact with third-party systems through REST.

ЗМІСТ

Вступ.....	6
1 Аналіз проблемної галузі та постановка задачі.....	7
1.1 Проблема багатокритеріального вибору	7
1.2 Аналіз схожих інструментів щодо визначення інноваційності продукту	9
1.3 Постановка задачі.....	22
2 Аналіз методів розв'язання задачі багатокритеріального вибору.....	24
2.1 Методи розв'язання задачі багатокритеріального вибору	24
2.2 Метод згортки.....	24
2.2.1 Аддитивна згортка	25
2.2.2 Мультиплікативна згортка	26
2.2.3 Максінна згортка	26
2.3 Метод граничних критеріїв	27
2.4 Метод головного критерію.....	27
2.5 Метод відстані	28
2.6 Побудова множини Парето	29
2.7 Пошук найкращого рішення при впливі навколишнього середовища.....	31
2.7.1 Метод гарантованого результату	31
2.7.2 Принцип Неша.....	32
2.8 Результати аналізу існуючих методів	33
3 Опис прийнятих проектних рішень.....	35
3.1 UML-моделювання системи	35
3.2 Розробка моделі системи визначення найкращої альтернативи	37
3.2.1 Нормалізація критеріїв	37
3.2.2 Визначення вагових коефіцієнтів.....	39
3.2.3 Використання методів згортки	40

3.2.4 Аналіз результатів згортки.....	43
3.3 Розробка архітектури застосунку	44
3.4 Опис програмних рішень	48
3.4.1 Вибір засобів розробки.....	49
3.4.2 Опис програмної системи	51
Висновки	55
Перелік джерел посилання	57
Додаток А.....	59

ВСТУП

Майже кожна людина подорожує один чи декілька разів на рік. Вона витрачає час на те, щоб спланувати подорож. Для цього потрібно знайти літак, забронювати готель або ж квартиру, дослідити заклади, де можна харчуватися. Також потрібно вивчити, які туристичні місця найпопулярніші та їх вартість.

Планування займає великий час, бо дуже багато критеріїв потрібно визначити, щоб обрати готель, літак (наприклад, якісь послуги обов'язкові), ресторан тощо. В процесі рішення людина може змінювати критерії за ненааявністю тієї чи іншої послуги, довго шукати найоптимальніші ціни на літак, зважаючи на місцезнаходження готелю та різні ризики, пов'язані з заторами та погодними явищами у місті, обрати інший готель. Враховуючи ці фактори, дуже корисною стане система, що на основі введених критеріїв різних пріоритетів пропонуватиме користувачеві оптимальний план подорожі, включаючи авіап перевезення, готель, обід і вечерю, та навіть різноманітні культурні заходи, враховуючи зазначені вище проблеми та ризики.

Сьогодні існує велика кількість продуктів, що стосуються подорожей. Багато з них можуть надати послуги по якимось напрямам, або навіть по декількох. Але жоден не може запропонувати найоптимальніший для користувача повний план подорожі, засновуючись на багатьох критеріях різного пріоритету. Це дуже буде потрібне особам, що вирушають у бізнес-тріп, і такі послуги стали б у нагоді, коли майже увесь час має бути розпланованим. Отже, розглядаючи проблемну галузь, слід відзначити, що обрана тема є актуальною на сьогодні і в близькому майбутньому. Щоб проаналізувати дану предметну галузь треба сказати про існуючі програмні рішення, що використовуються, на базі яких буде побудована нова система, їх плюси та мінуси.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Проблема багатокритеріального вибору

Слово критерій перекладається з грецької як засіб судження. Його наукова сутність — це мірило для оцінювання предмета (явища) чи ознака, взята за основу класифікації.

Нехай особою прийняття рішень обрано певну ціль щодо розвитку економічної системи, множину альтернативних стратегій (рішень), реалізація яких дає змогу досягти поставленої цілі, а також низку критеріїв оцінювання цих стратегій (в якості критеріїв можуть використовуватися спеціалізовані економічні показники). Науковий аналіз проблем прийняття рішень в економічній діяльності та підприємстві починається з моменту, коли хоча б частина альтернатив і (чи) критеріїв відома. У сучасній науці стосовно прийняття рішень центральне місце посідають багатокритеріальні задачі вибору. Вважається, що врахування багатьох критеріїв наближує постановку задачі до реального життя.

Наприклад, особа прийняття рішень обирає рейс для подорожі. У якості альтернатив було обрано декілька рейсів і виокремлено наступні критерії вибору: клас, наявність WI-FI, тип авіалінії та кількість пересадок.

Зазначимо, що традиційно заведено розрізняти три основних класи задач прийняття рішень:

- упорядкування альтернатив. Для низки задач вважається досить обґрунтованою вимога щодо впорядкування стратегій (об'єктів) на множині альтернатив. Так, наприклад, члени родини впорядковують за ступенем необхідності майбутні витрати на придбання певних товарів і послуг, мандрівники упорядковують за рейтингом, відстанню від центру, наявністю ресторану та багатьма іншими критеріями готелі для ночування під час подорожі тощо. У загальному випадку вимога

щодо впорядкування альтернатив означає необхідність визначити відносну цінність кожної з них;

– розподіл альтернатив за класами рішень. Такі задачі зустрічаються у повсякденному житті, наприклад, у разі купівлі квартири чи будинку. В разі пошуку квитка на літак мандрівники, як правило, розділяють альтернативи на дві групи: такі, що заслуговують і не заслуговують більш детального вивчення. Групи товарів розрізняються за якістю тощо;

– виокремлення (вибір) кращої альтернативи. Ця задача традиційно вважається однією з основних у прийнятті рішень. Вона часто зустрічається на практиці. Вибір ресторану для харчування під час подорожі, вибір місця праці, вибір інвестиційного проекту з множини альтернативних варіантів — це типові приклади.

Наголосимо, що багатокритеріальні задачі — це задачі, обтяжені невизначеністю, конфліктністю та породженим ними ризиком. Частина інформації, необхідна для вичерпного та однозначного визначення вимог до рішення, принципово відсутня на момент прийняття цього рішення. Дослідник відносно часто, в принципі, вербально може визначити основні змінні, враховуючи зв'язки (залежності) між ними, тобто побудувати концептуальну модель, що відображає ситуацію. Але залежності між критеріями взагалі не можуть бути визначені на ґрунті лише об'єктивної інформації, що є в розпорядженні дослідника. Такі проблеми, згідно з класифікацією, запропонованою в 1958 р. у статті Г.Саймона та А.Ньюелла, слід віднести до слабоструктурованих, оскільки тут нестача об'єктивної інформації є принципово неусуненою на момент прийняття рішення, тобто об'єктивно існує невизначеність і породжений нею ризик.

Існує низка проблем, у котрих відомий лише перелік основних параметрів, але кількісні взаємозв'язки між ними встановити неможливо (відсутня необхідна інформація). Інколи ясно лише, що зміна параметрів у певних межах впливає на рішення. У таких випадках структура, що розуміється як множина зв'язків між параметрами, не визначена і проблема називається неструктурованою. Типовими

неструктурованими проблемами є вибір професії, вибір місця роботи тощо. Усі ці проблеми потребують урахування невизначеності та моделювання ризику, кількісна оцінка якого, у свою чергу, теж є багатокритеріальною.

1.2 Аналіз схожих інструментів щодо визначення інноваційності продукту

У процесі дослідження виявлено декілька масштабних продуктів, відомих усьому світу. Далі буде розглянуто плюси та мінуси цих додатків, історію розвитку, їх особливості та фінансові успіхи, щоб зрозуміти, як само їх інтегрувати, аби пропонувати найоптимальніше рішення засновуючись на групі різнопріоритетних критеріїв, та що можливо покращити, аби зробити новий продукт ще успішніше за попередників з фінансової точки зору.

Zomato (<https://www.zomato.com/>) – індійська служба пошуку та бронювання ресторанів (див. рис.1.1), що заснована в 2008 році компанією Deepinder Goyal та Rankaj Chaddah. В даний час вона працює в 24 країнах. Вона надає інформацію та огляди ресторанів, а також послуги онлайн-доставки та бронювання столів в деяких країнах.

Платформа пошуку та відкриття ресторану почала свою діяльність під назвою Foodiebay. У листопаді 2010 року бренд було перейменовано у Zomato.

У вересні 2012 року компанія Zomato розширила зарубіжні країни до Об'єднаних Арабських Еміратів, Шрі-Ланки, Катару, Великобританії, Філіппін та Південної Африки. У 2013 році компанія запустила продукт в Новій Зеландії, Туреччині, Бразилії та Індонезії з її веб-сайтом і додатками, доступними на турецькій, бразильській, португальській, індонезійській та англійській мовах.

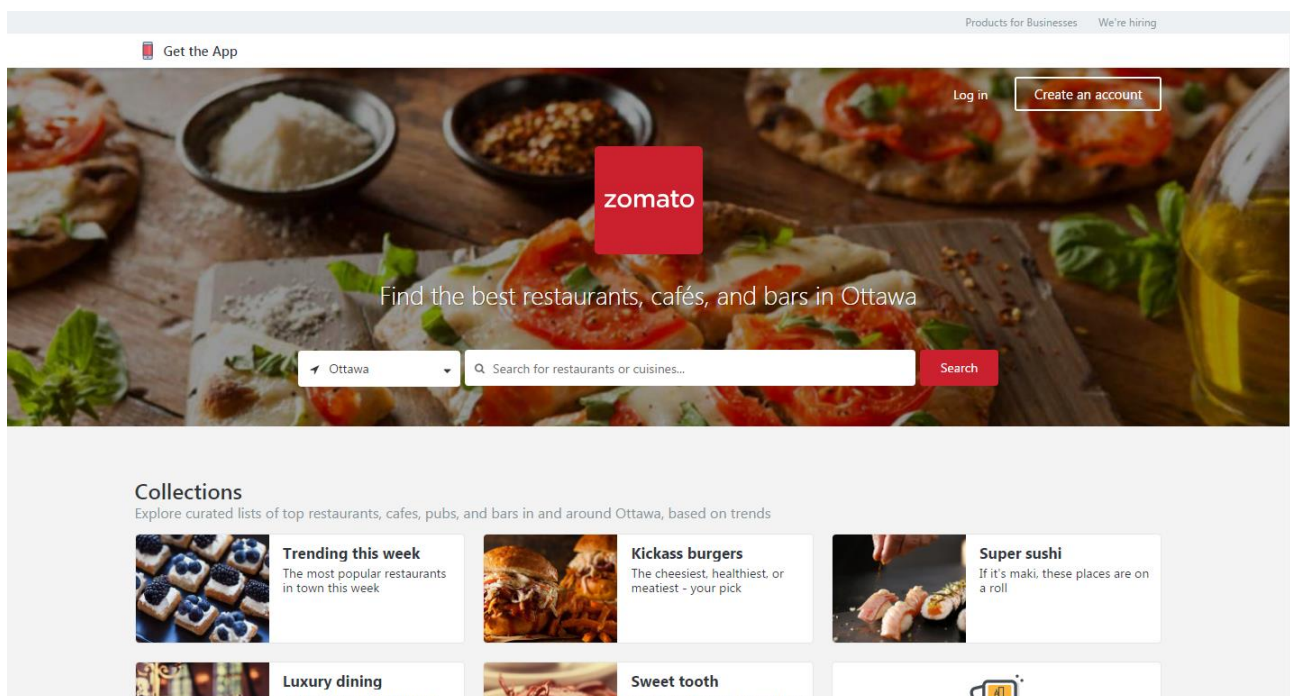


Рисунок 1.1 – Головна сторінка Zomato

У квітні 2014 року компанія Zomato розпочала надавати свої послуги в Португалії, а потім у Канаді, в Лівані та Ірландії.

Придбання заснованого на Сіетлі продовольчого порталу Urbanspoon відзначило вступ фірми до Сполучених Штатів, Канади та Австралії і привело його до прямої конкуренції з Yelp, Zagat та OpenTable.

У вересні 2017 року Zomato заявила, що компанія "стала прибутковою" в 24 країнах, в яких вона в даний час працює. Крім того, Zomato оголосила, що "модель нульової комісії" буде запроваджена для ресторанів-партнерів.

Між 2010 та 2013 роками Zomato підняв приблизно 16,7 млн доларів США з Info Edge India, даючи їм 57,9% пакета у Zomato. У листопаді 2013 року Zomato зібрав додаткові 37 млн доларів США від Sequoia Capital та Info Edge India.

У листопаді 2014 року Zomato завершила черговий раунд фінансування 60 млн доларів США після оцінки грошей у розмірі 660 млн доларів США. Цей раунд фінансування спільно очолював Info Edge India та Vy Capital, разом з Sequoia Capital.

У квітні 2015 року компанія Info Edge India, Vy Capital та Sequoia Capital провели ще один раунд фінансування на суму 50 млн доларів США. Далі було виділено ще 60 млн доларів США фінансування на чолі з Temasek, урядовою інвестиційною компанією у Сінгапурі, разом з Vy Capital у вересні.

У жовтні 2018 р. Компанія Zomato зібрала 210 млн доларів США з оплати філії Алібаби Ant Financial. Ant Financial отримала частку у власності більше 10% компанії в рамках туру, який оцінив Zomato у розмірі близько 2 млрд доларів США. Zomato також підняв додаткові 150 млн доларів США також від Ant Financial раніше 2018 року.

Використовуючи цю систему можна робити пошук за допомогою фільтру за багатьма критеріями. Наприклад, існує можливість фільтрувати заклади за популярністю, типом страви або типом самого закладу (рис.1.2).

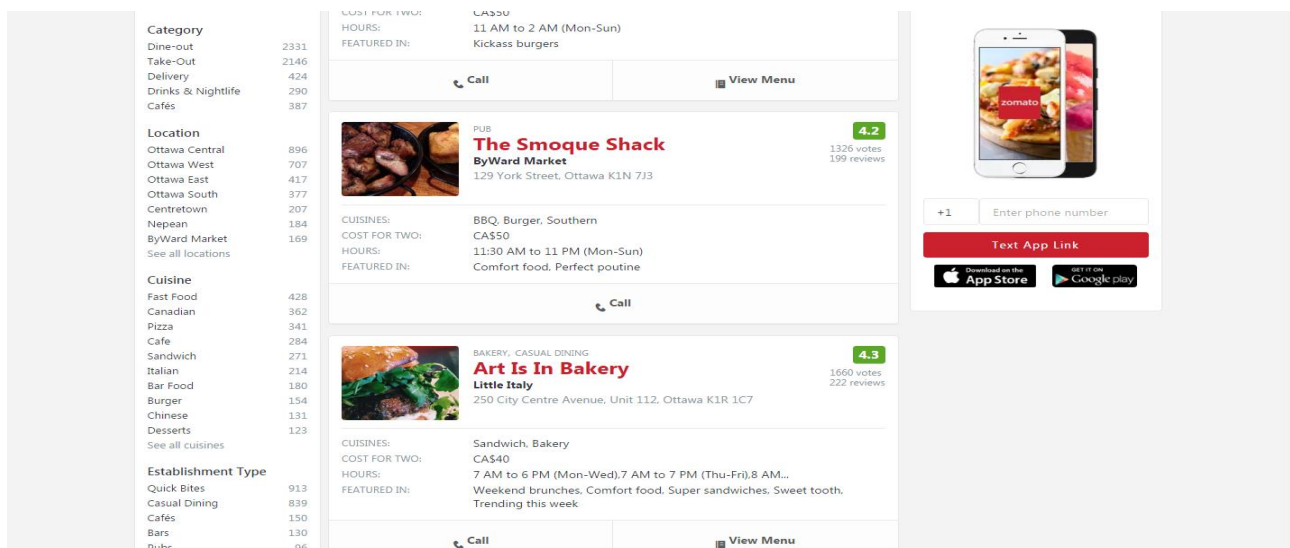


Рисунок 1.2 – Можливість фільтрації закладів за типом страви Zomato

Для кожного закладу можна подивитися будь-яку інформацію, що може бути використана при плануванні подорожі, така як часи роботи, контактний номер телефону чи місцезнаходження, середня ціна обіду, тип кухні тощо (див. рис.1.3).

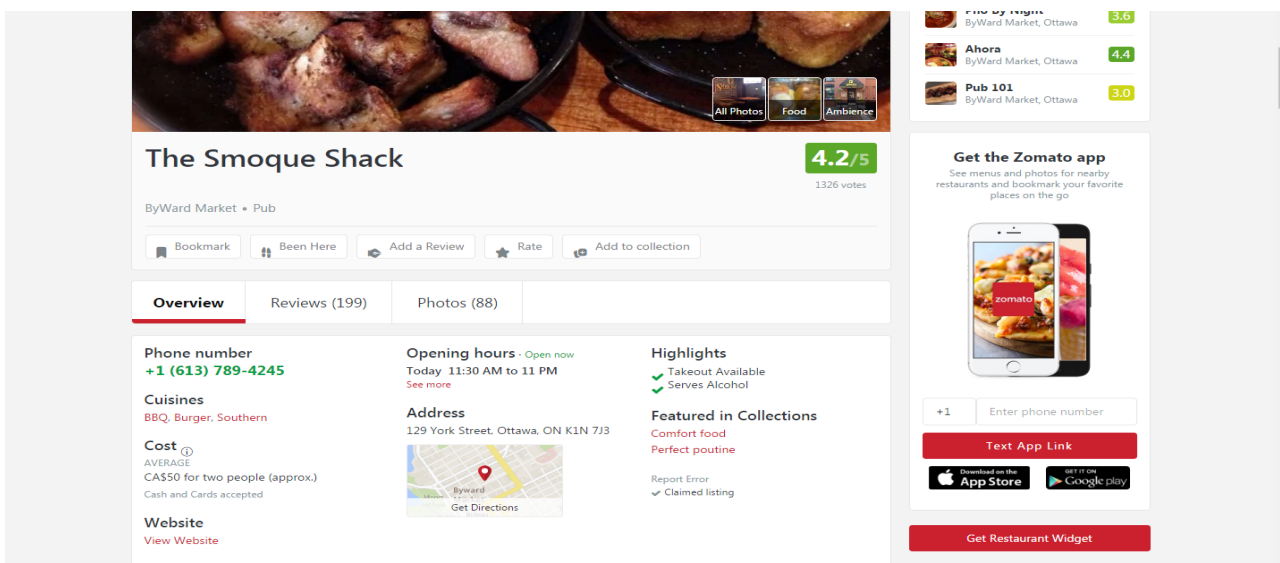


Рисунок 1.3 – Сторінка інформації про заклад Zomato

Можливо, головним плюсом цієї системи є можливість онлайн-бронювання стола у будь-якому закладі, що теж дуже зручно для планування подорожі (рис.1.4).

Рисунок 1.4 – Бронювання стола Zomato

Згідно з вищеописаним, Zomato — дуже корисна система, яка повністю сконцентрована на послугах харчування, і її успіхи на світовому ринку не дивують. Zomato — саме той продукт, інтеграція з яким може запропонувати дуже якісне рішення проблеми багатокритеріального вибору закладу для харчування, як частини плану подорожі.

Але, звичайно, є й недоліки. Адже Zomato поширений в обмеженій кількості країн. Саме тому для повноти функціональності може бути знайдено ще декілька додатків, що спеціалізуються на харчуванні, тоді планувати подорож можливо буде більш повно стосовно розкладу.

Skyscanner (<https://www.skyscanner.com.ua/>) – система веб-пошуку інформації про повітряні перевезення (рис.1.5), яка дозволяє користувачам переглядати інформацію про авіарейси за різноманітними критеріями. Працює на 30 мовах. Штаб-квартира цього сервісу розташовується в Единбурзі, Шотландія.

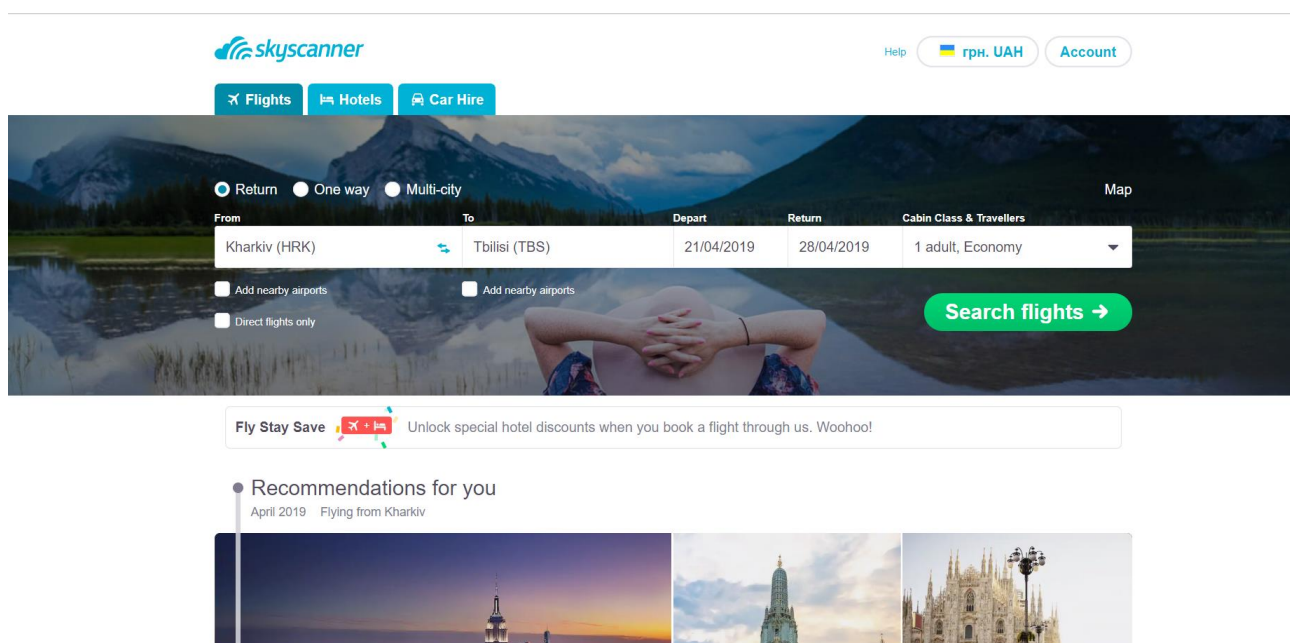


Рисунок 1.5 – Головна сторінка Skyscanner

Компанія була заснована в 2003 році трьома ІТ фахівцями: Гарефом Вільямсом, Беррі Смітом, і Бонамі Граймс, після того, як великому любителю катання на лижах,

Гарефу Вільямсу, набридли труднощі в пошуку дешевих авіаквитків на гірськолижні курорти.

Додаткові можливості, такі як графіки тарифів, дозволяють користувачам порівнювати ціни на авіаквитки по будь-якому обраному напрямку протягом місяця. Можна також дізнатися тарифи на вихідні за всіма доступними напрямками, задавши тільки конкретну точку вильоту.

Веб-сайт компанії доступний на 30 мовах, включаючи російську, іспанську, китайську, французьку, італійську, німецьку, українську, португальську, а також японську.

Skyscanner не займається продажем авіаквитків безпосередньо, але веб-сайт може бути використаний для того, щоб знайти найбільш дешевий варіант для бажаного напрямку, після чого користувача буде автоматично переадресовано на сайт відповідної авіакомпанії або турагентства для створення самого бронювання.

Користувачі можуть здійснювати пошук без необхідності введення конкретних дат місць призначення, що робить цю систему дуже гнучкою. Унікальна технологія індексування даних, розроблена компанією, дозволяє системі Skyscanner проводити пошук набагато швидше мета-пошукових систем, оскільки відпадає потреба в зборі інформації з інших веб-сайтів безпосередньо при кожному пошуку. Це яскраво відрізняє систему Skyscanner від більшості інших сайтів з пошуку і бронювання авіаперевезень.

В даний час веб-сайт Skyscanner відвідує понад 60 мільйонів осіб щомісяця. У 2016 році система обробляла понад 3 мільярдів запитів кожного дня.

Сайт був включений в «десятку найкращих туристичних сайтів» (газета «Independent») і в список «101 дуже корисний веб-сайт». Skyscanner був удостоєний різноманітних нагород, включаючи «найкращий сайт 2005 року в розділі інтернет-технологій» (портал «TravelMole») і срібну нагороду журналу «Wanderlust» в розділі найкращих туристичних сайтів 2005 року.

Також Skyscanner дозволяє робити аналогічний пошук апартаментів та оренду автомобілю на час подорожі (рис.1.6).

Рисунок 1.6 – Сторінка пошуку готелів Skyscanner

Як можна зрозуміти, Skyscanner не дарма є найпопулярнішою системою пошуку і бронювання авіаперевезень, і його досягнення не є дивовижними. Його API може надати увесь необхідний функціонал для інтеграції та сприяти пошуку авіаперевезень на основі багатьох критеріїв, що даватиме змогу реалізувати частину плану подорожі, що стосується перельоту.

Звичайно, Skyscanner – найпопулярніший додаток для пошуку й купування квитків на літак, але недоліком його, порівняно з нашою майбутньою системою, є те, що обрати літак, та навіть готель, з бажаними умовами можливо тільки фільтрацією, не враховуючи пріоритети користувача. Та й ще потрібно переглядати усю безліч варіантів, що залишились, та й обрати оптимальне для себе рішення. Також не має змоги зробити повний план подорожі, враховуючи і літак, і готель.

TripAdvisor (<https://www.tripadvisor.com/>) – це веб-сайт для подорожей і ресторанів (див. рис.1.7), який показує огляди готелів, ресторанів, бронювання

номерів та інший зміст, пов'язаний з подорожами. Він також включає в себе інтерактивні форуми подорожей.

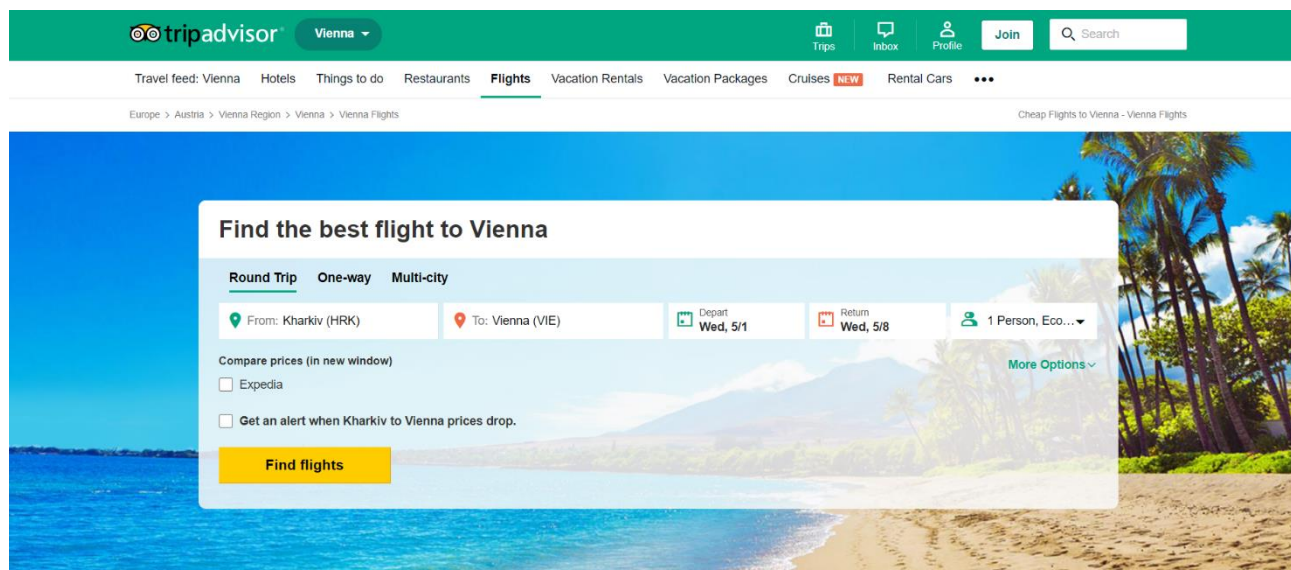


Рисунок 1.7 – Головна сторінка TripAdvisor

Маючи штаб-квартиру в місті Нідхем, штат Массачусетс, TripAdvisor став першим користувачем контенту, створеного користувачем. Послуги веб-сайту безкоштовні для користувачів, які надають більшу частину вмісту, а веб-сайт насамперед підтримується об'єктом бронювання готелів та рекламною бізнес-моделлю.

TripAdvisor було засновано Стефаном Кауфером, Ленглі Стейнертом, Ніком Шанні і Томасом Палкою у лютому 2000 року. Кауфер заявив, що оригінальна ідея не була створена користувачем сайтів соціальних медіа, щоб обмінятися відгуками: "Ми почали як сайт, де ми більше орієнтувалися на ті офіційні слова з путівників, газет або журналів." Оригінальне фінансування було отримано від компаній Flagship Ventures, Bollard Group і приватних інвесторів.

У 2004 році компанію придбала IAC / InterActiveCorp.

У серпні 2005 року IAC випустила свою туристичну групу підприємств під назвою Expedia, Inc.

У квітні 2009 року TripAdvisor запустив свій офіційний сайт у Китаї. З того часу TripAdvisor індексував інформацію про більше, ніж 20 тисяч готелів та ресторанів (рис.1.8), та про відгуки клієнтів, а також зробив перші списки, ставши одним з найбільших туристичних сайтів у світі станом на липень 2011 року.

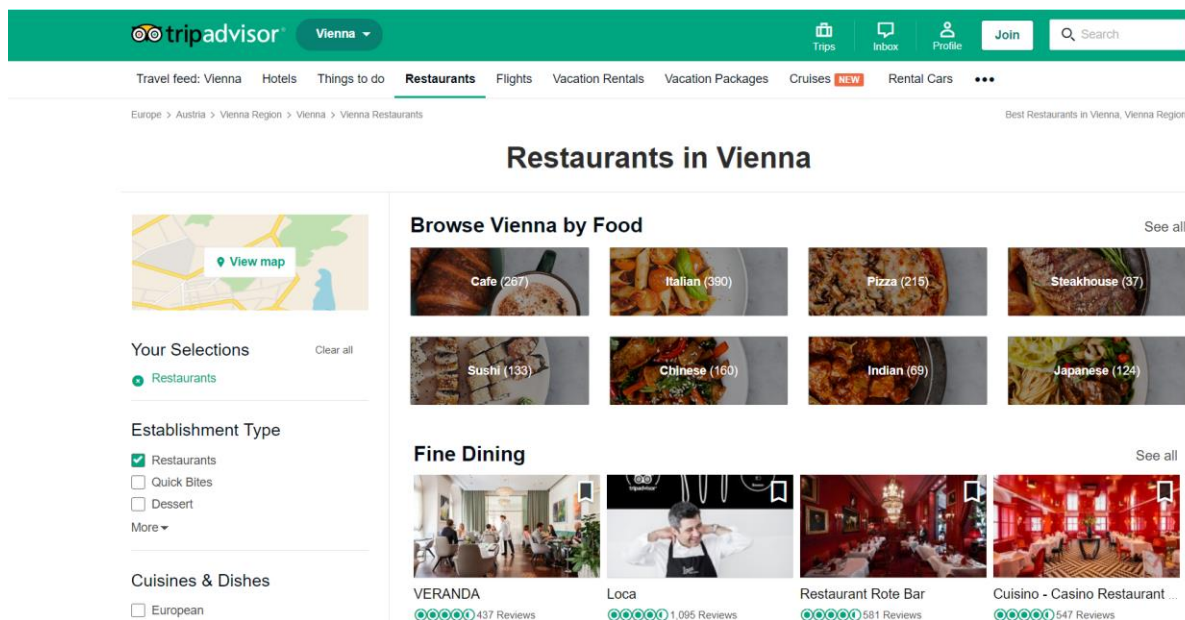


Рисунок 1.8 – Сторінка пошуку ресторанів TripAdvisor

У вересні 2010 року SmarterTravel, що входить до складу медіа-групи TripAdvisor, запустив SniqueAway (тепер Jetsetter), перший сайт, де тільки кожен учасник подорожі схвалюється людьми.

У квітні 2011 року було оголошено, що Expedia розділиться на дві компанії, перевернувши бренд туристичних сайтів у TripAdvisor. За словами генерального директора Expedia Дари Хосровшахі, цей крок "дозволяє двом підприємствам бути чистими і грати належним чином для того, щоб зростати відповідно".

Згідно з опитуванням PhoCusWright, проведеним у липні 2011 року, 3641 респондент, який отримав довільну заявку через спливаюче посилання на TripAdvisor.com і замовлений Trip Advisor, «98% учасників виявили, що відгуки готелів TripAdvisor точно відображають досвід».

TripAdvisor стає найбільшим у світі туристичним сайтом з майже 280 мільйонами унікальних щомісячних відвідувачів.

У квітні 2012 року компанія запустила підключення до Facebook, яке дозволяє користувачам фільтрувати відгуки від людей з їх власного соціального графіку.

У жовтні 2014 року Trip Advisor опублікував нову функцію "Просто для Вас", яка пропонує рекомендації готелів, що базуються на налаштуваннях користувача та історії пошуку на сайті. Ця функція віддзеркалювала функцію "Племена", що була запущена на сайті конкурента Gogobot в 2013 році.

У листопаді 2018 року компанія включила в свій ресторанний список замовлення продуктів і послуг доставки – DoorDash.

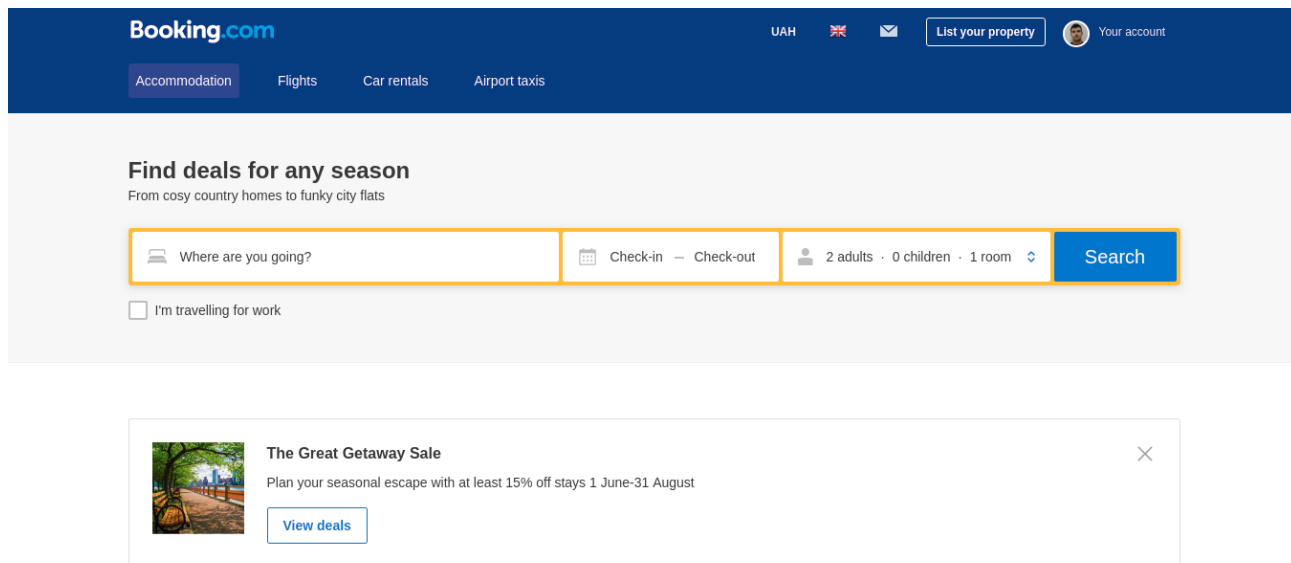
Відповідно до щорічного звіту акціонерів, TripAdvisor заробляє 43% своїх доходів через своїх двох найважливіших рекламних партнерів, Expedia і Priceline (і їх дочірніх компаній). Вона отримує значну частину свого доходу від реклами та переадресацій. Вони включають доходи від продажу реклами, переважно через рекламу на основі кліків і, в меншій мірі, рекламу на основі дисплея та підписки. Компанія також має головні рекламні контракти зі своїми рекламними партнерами.

Як ми бачимо, TripAdvisor — досить універсальна система, яка дозволяє користувачу знайти, все, що стосується подорожі, але окремо обираючи готель, ресторан тощо та за допомогою різних фільтрів. Нова система зможе, так би мовити, пропонувати об'єднані послуги цього продукту з урахуванням багатьох критеріїв.

Звичайно, TripAdvisor – універсальний додаток для пошуку й купування квитків на літак, бронювання готелю та пошуку ресторанів, але недоліком його, порівняно з нашою майбутньою системою, є неможливість зробити повний план подорожі, враховуючи літак, готель, ресторан тощо. Також попередні додатки спеціалізуються на якійсь конкретній частині (готель, літак, ресторан тощо), а саме тому мають більше функціональності щодо окремої частини.

Booking.com (<https://www.booking.com/>) є веб-сайтом агрегатора проїзних тарифів і пошуковим механізмом для бронювання (див. рис.1.9). Цей продукт є

власністю та керується і є основним джерелом доходу в базованому в США Booking Holdings. Штаб-квартира Booking.com знаходиться в місті Амстердам.



The screenshot shows the Booking.com homepage. At the top is a dark blue navigation bar with the Booking.com logo, currency (UAH), flags, a mail icon, a 'List your property' button, and a user account icon labeled 'Your account'. Below this is a light blue bar with links for 'Accommodation', 'Flights', 'Car rentals', and 'Airport taxis'. The main section is titled 'Find deals for any season' with the subtitle 'From cosy country homes to funky city flats'. It features a search bar with the placeholder 'Where are you going?', a date selector for 'Check-in' and 'Check-out', a guest selector for '2 adults · 0 children · 1 room', and a blue 'Search' button. Below the search bar is a checkbox for 'I'm travelling for work'. At the bottom, there is a promotional banner for 'The Great Getaway Sale' with a small image of a park, the text 'Plan your seasonal escape with at least 15% off stays 1 June-31 August', and a 'View deals' button.

Рисунок 1.9 – Головна сторінка Booking.com

На сайті розміщено 28,425,726 оголошень у 148,470 пунктах призначення у 228 країнах та територіях світу. Щодня на сайті зберігається понад 1,550 тисяч номерів. Сайт доступний на 43 мовах. Його гасло «Дома, будинки і все між ними», хоча в рекламних роликах це «Booking.com: Booking.yeah».

Booking.com було створено, коли bookings.nl, заснована в 1996 році Гертом-Ян Брюйнсмі, об'єдналася в 2000 році з Bookings Online, заснованою компанією Sisco та Alec Behrens, Marijn Muysen та Bas Lemmens, яка діяла як Bookings.org. Назва та URL були змінені на Booking.com, а Stef Noorden був призначений його генеральним директором. У 1997 році Bruinsma хотів розмістити оголошення в De Telegraaf, голландській газеті з найвищим тиражем. Оголошення було відхилено, оскільки De Telegraaf приймав оголошення лише з номером телефону, а не з веб-сайтом. У 2002 році Expedia відмовилася купувати bookings.nl.

У липні 2005 року компанія була придбана компанією Booking Holdings (або Priceline Group, як вона називалася на той час) за 133 млн доларів США, а пізніше вона

співпрацювала з Європейською онлайн-компанією з бронювання готелів ActiveHotels.com, що була придбана компанією Booking Holdings / Priceline за 161 млн доларів США.

У 2006 році компанія Active Hotels Limited офіційно змінила назву на Booking.com Limited. Інтеграція успішно допомогла своїм попередникам поліпшити своє фінансове становище від втрати в 19 млн доларів США у 2002 році до 1,1 млрд доларів США прибутку в 2011 році. Ринок подорожей виявився настільки прибутковим.

У серпні 2012 року китайська туристична компанія Ctrip створила партнерство з компанією, яка дозволить Ctrip отримати доступ до глобального портфоліо Booking.com. Партнерство було поглиблене пізніше у червні 2018 року.

У лютому 2013 року група компаній "Панорама", найбільша індонезійська туристична компанія, сформувала стратегічне партнерство з Booking.com, щоб отримати доступ до портфоліо готелів.

У 2014 році Sprylogics International Corporation, яка надає місцевим мобільне рішення для споживачів, а також мобільні додатки, підписала угоду з Booking.com, і тому Sprylogics Poynt App і Poynt-Enabled SDK зможуть скористатися обширними даними про готель Booking.com.

У жовтні 2014 року Ural Airlines створили партнерство з Booking.com.

Компанія Booking.com та її дочірні компанії були найпопулярнішими у сфері подорожей та туризму для Google Adwords у 2016 році, витративши 3,5 млрд доларів США.

У січні 2013 року перша кампанія бренду Booking.com, "Booking.yeah", була запущена в режимі онлайн, розміщена на телевізійних станціях і в кінотеатрах і на телевізійних мережах, для ринку США з рекламним агентством Wieden + Kennedy Amsterdam. У вересні 2013 року Австралія стала другим ринком для перегляду кампанії. У січні 2014 року компанія розпочала рекламну кампанію в Канаді, в лютому

2014 року компанія розпочала рекламну кампанію в Великобританії. У липні 2014 року компанія розпочала рекламну кампанію в Німеччині.

У листопаді 2010 року компанія запустила мобільний додаток для готелів та житлових приміщень для iPad.

У лютому 2011 року компанія запустила свою мобільну програму на Android.

У жовтні 2012 року компанія запустила свій перший додаток для Microsoft Windows, використовуючи Windows 8.

У грудні 2012 року компанія запустила програму Kindle Fire, яку можна завантажити у всіх магазинах Amazon, включаючи такі країни як США, Великобританію, Німеччину, Іспанію, Францію, Італію, Японію та багато інших.

Booking.com дозволяє обрати серед вільних кімнат готелю, ту, яка найбільше сподобається користувачеві за бажаними характеристиками (рис.1.10).

Room type	Sleeps	Price for 15 nights	Your choices	Select rooms	
Classic Room 1 single bed Air conditioning Private bathroom • Desk • Electric kettle • Refrigerator • Hairdryer • Safety Deposit Box • Cable Channels • Satellite Channels More Prices are per room for 15 nights		UAH 66,489 taxes and charges may vary	Smart price The budget-friendly choice Book and pay securely with Booking.com • This reservation will not count towards your Genius status • Invoice not provided with this option • Non-refundable	0 ▾	I'll reserve It only takes 2 minutes 7 other people looked for your dates in the last 10 minutes
Junior Suite Only 1 left on our site! 1 large double bed 40 m² Flat-screen TV Air conditioning Private bathroom Free WiFi • View • Telephone • Satellite Channels • Cable Channels • Laptop safe • Safety Deposit Box • Trouser Press • Seating Area • Heating • Wardrobe or closet • Hairdryer		UAH 81,135 includes taxes and charges	Very good breakfast UAH 563 • Non-refundable	0 ▾	
		UAH 85,363 includes taxes and charges	Very good breakfast UAH 563 Partially refundable NO PREPAYMENT NEEDED - pay at the property	0 ▾	
		UAH 89,856 includes taxes and charges	Very good breakfast UAH 563 FREE cancellation before 18:00 on 30 May 2019	0 ▾	

Рисунок 1.10 – Сторінка обрання номеру Booking.com

Booking.com — найпопулярніший додаток для бронювання готелів, що сконцентрований саме на цій функціональності. Він пропонує API, що може бути використаний для інтеграції та вирішення багатокритеріальної задачі пошуку готелів, як частини планування подорожі.

Недоліком цього додатку є те, що за-для того, щоб обрати для себе готель, потрібно витратити багато часу на вивчення пропонованих варіантів, продивитися сторінки вибору кімнат, що різняться за сервісами та цінами. Нова система матиме можливість запропонувати користувачеві обрати критерії з пріоритетами, за якими йому буде запропонована кімната без вивчення інших варіантів, які все одно відкинуться. Також цей сервіс має лише можливість обрати готель, а не створити повний план подорожі, як це є задачею нашої нової системи.

1.3 Постановка задачі

Таким чином, на підставі проведеного порівняльного аналізу сервісів з надання послуг, що стосуються планування подорожей, слід зробити наступні висновки:

- більшість сервісів здійснюють консолідацію можливих планів подорожей на підставі переваг фінансових вкладів спонсорів, ранжуючи отримані результати згідно них;
- результати планування не зберігаються;
- кожен з додатків пропонує можливість планування лише однієї частини подорожі (літак, готель тощо).

Виходячи з вище наведених недоліків існуючих аналогів, можливо зазначити актуальність роботи, яка полягає в тому, що, на відміну від проаналізованих додатків, новий програмний продукт матиме змогу пропонувати користувачеві повний план подорожі, включаючи квитки на літак, готель тощо, зважаючи на важливі для

користувача критерії кращої подорожі та їх пріоритети. І саме питання обробки цих критеріїв дає змогу сформулювати мету даної роботи.

Метою роботи є розробка методу ранжування альтернатив багатокритеріального вибору з використанням математичної моделі та здійснення програмної реалізації математичної моделі за-для доведення результатів планованого експерименту.

Для цього потрібно проаналізувати існуючі методи вирішення задачі багатокритеріального вибору, визначити, які з них найкраще підійдуть для умов даної задачі побудування системи, та сформулювати алгоритм впорядкування альтернатив, користуючись цими методами.

2 АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ БАГАТОКРИТЕРІАЛЬНОГО ВИБОРУ

2.1 Методи розв'язання задачі багатокритеріального вибору

Багатокритеріальна задача вибору [1] формулюється в наступному вигляді. Дано множину допустимих альтернатив, кожна з яких оцінюється множиною критеріїв. Потрібно визначити найкращу альтернативу. При її вирішенні основна складність полягає в неоднозначності вибору найкращого рішення. Для її усунення використовуються дві групи методів. У методах першої групи прагнуть скоротити число критеріїв, для чого вводять додаткові припущення, що відносяться до процедури ранжування критеріїв і порівняння альтернатив. У методах другої групи прагнуть скоротити число альтернатив у вихідній множині, виключивши свідомо погані альтернативи.

До методів першої групи відносяться метод згортки, метод головного критерію, метод граничних критеріїв, метод відстані. Слід зазначити, що суворе обґрунтування цих методів відсутнє і їх застосування визначається умовами завдання і перевагою особи прийняття рішень.

2.2 Метод згортки

Метод згортки [2] полягає в заміні вихідних критеріїв (їх називають також локальними або частковими) K_j одним загальним критерієм K . Ця операція називається згорткою або агрегуванням приватних критеріїв. Метод доцільно застосовувати, якщо за умовами завдання часткові критерії можна розташувати по спадаючій важливості так, що важливість кожної пари сусідніх критеріїв

розрізняється не сильно, або, якщо альтернативи мають істотно різні оцінки за різними критеріями. Найбільш часто використовуються аддитивна, мультиплікативна та максимінна згортки.

2.2.1 Аддитивна згортка

Аддитивна згортка [3] має наступний вигляд:

$$K(x) = \sum_{j=1}^n a_j K_j(x), \quad (2.1)$$

де $K(x)$ – загальний критерій для альтернативи $x \in X$, який показує її придатність для досягнення мети;

$(K_1(x), \dots, K_j(x), \dots, K_n(x))$ – набір вихідних критеріїв;

n – число вихідних критеріїв;

a_j – відносна вага (важливість) часткового критерію K_j . Для ваг виконується умова нормування $\sum_{j=1}^n a_j = 1$, яка необхідна, щоб результати, отримані в різних умовах, можна було порівняти.

Найкраще рішення визначається виразом:

$$x^* = \arg \max_{x \in X} K(x). \quad (2.2)$$

Тобто найкращим вважається рішення, якому відповідає максимум загального критерію на множині альтернатив.

2.2.2 Мультиплікативна згортка

Мультиплікативна згортка [4] має наступний вигляд:

$$K(x) = \prod_{j=1}^n K_j^{a_j}(x). \quad (2.3)$$

Найкраще рішення визначається виразом (2.2).

2.2.3 Максінна згортка

Максінна згортка [5] має вигляд:

$$K(x) = \min_j a_j K_j(x). \quad (2.4)$$

Ця згортка враховує критерій, який має найменше значення. Іноді при її застосуванні вважають, що ваги критеріїв близькі один до одного, або всі критерії мають однакову важливість, тобто $a_j = \text{const}(j) = \frac{1}{n}$. В цьому випадку вона називається згорткою без урахування ваги критеріїв і тому набирає наступного вигляду:

$$K(x) = \min_j K_j(x). \quad (2.5)$$

Найкраще рішення визначається виразом (2.2). Підставивши в (2.2) вираз (2.4), отримаємо:

$$x^* = \arg \max_{x \in X} \min_j a_j K_j, \quad (2.6)$$

тому цю згортку називають по основних операціях максимінною. Використання того чи іншого типу згортки відображає уявлення особи прийняття рішень про стратегію.

2.3 Метод граничних критеріїв

Метод граничних критеріїв [6] часто застосовується в задачах забезпечення, наприклад при плануванні і проектуванні, коли обмеження задаються у вигляді:

$$K_j(x) \geq K_{j_0}; j = 1, \dots, n, \quad (2.7)$$

де $K_{j_0}(x)$ – порогові значення критеріїв. Їх сукупність зазвичай характеризує деякий аналог (базовий рівень).

Утворюємо згортку:

$$K(x) = \min_j \left(\frac{K_j(x)}{K_{j_0}} \right). \quad (2.8)$$

Найкраще рішення визначається виразом вигляду (2.2).

2.4 Метод головного критерію

Метод головного критерію [7] полягає в заміні багатокритеріальної задачі однокритеріальною із додатковими обмеженнями. Цей метод не вимагає нормалізації

критеріїв і кількісного задання їх пріоритетів, але необхідно мати інформацію про порогові значення неголовних критеріїв.

Якщо вихідної інформації достатньо, щоб з множини вихідних критеріїв $K_j(x)$ виділити головний (основний) $K_0(x)$, тобто такий, що значно перевершує за важливістю всі інші критерії, то найкраще рішення визначається у вигляді:

$$x^* = \arg \max_{x \in X} K_0(x), \quad (2.9)$$

при додаткових умовах $K_j(x) \geq K_{j_{\text{порог}}}$ для всіх інших критеріїв, тобто їх значення повинні бути не менше деяких порогових значень.

2.5 Метод відстані

Метод відстані [8] полягає у введенні деякої метрики (відстані) в просторі критеріїв. Нехай вихідної інформації достатньо для того, щоб визначити "ідеальне" (еталонне) рішення, відповідне точці абсолютного максимуму в просторі критеріїв. Позначимо її, як $x_0(K_{0_1}, \dots, K_{0_n})$. Відзначимо, що ідеальне рішення на практиці не можна досягти і визначається лише теоретично. Введемо для кожної альтернативи $x \in X$ відстань до точки абсолютного максимуму $d(x)$. Найкраще рішення визначається як найбільш близьке до ідеального:

$$x^* = \arg \min_{x \in X} d(x). \quad (2.10)$$

В якості метрики відстані використовуються різні функції, наприклад, Махаланобіса, Мінковського. При використанні функції Мінковського:

$$d(x) = \left(\sum_{j=1}^n a_j |K_j(x) - K_{0j}|^p \right)^{\frac{1}{p}}, \quad (2.11)$$

при $p = 1$ отримуємо відстань Хеммінга;

$p = 2$ – евклідову відстань. Вибір параметру p залежить від умов завдання і переваг особи прийняття рішень.

Відзначимо, що якщо в якості ідеального рішення використовувати не абсолютний максимум, а абсолютний мінімум, то у виразі (2.10) операція *min* зміниться на операцію *max*.

2.6 Побудова множини Парето

Поряд з методами першої групи, які використовують згортку в просторі критеріїв, застосовуються методи другої групи, засновані на звуженні множини альтернатив, в яких намагаються зменшити число можливих варіантів рішень, виключивши свідомо погані. Один з підходів, що володіє великою спільністю, був запропонований італійським економістом В. Парето в 1904 році і називається методом, заснованим на принципі Парето або, коротко, методом Парето. Він застосовується, коли число альтернатив велике і альтернативи мають суперечливі оцінки за різними критеріями. У цьому випадку застосування методів першої групи може привести до ненадійних рішень і необхідний неформальний аналіз множини альтернатив. Для зменшення числа альтернатив вихідної множини будують множину Парето [9], що є підмножиною вихідної. Визначимо множину Парето:

$$x_p = \left(\begin{array}{c} x_p \in X \\ \exists x \in X \rightarrow \forall K_i (x_p) \geq K_i(x) \& \exists j \rightarrow K_j(x_p) > K_j(x) \end{array} \right). \quad (2.12)$$

Тобто альтернатива належить множині Парето, якщо вона не гірше за інших за всіма критеріями і хоча б за одним критерієм краще.

Альтернативи з множини Парето називаються парето-рішеннями, ефективними альтернативами. При вирішенні багатокритеріальних задач використовується принцип Парето, що полягає в тому, що найкраще рішення слід вибирати серед альтернатив, що належать множині Парето. Цей принцип виконується в більшості практичних ситуацій, коли альтернативи оцінюються по суперечливим критеріям. Він дозволяє звужити вихідну безліч альтернатив, причому остаточний вибір залишається за особою прийняття рішень. Альтернативи, що входять в множину Парето, попарно непорівнянні один з одним, тобто за одними критеріями краще одна альтернатива, за іншими інша і тощо, і їх неможливо поліпшити одночасно за всіма критеріями. Тому аналіз множини Парето дозволяє знайти компроміс між суперечливими вимогами і дає особі прийняття рішень можливість судити про те, яка "ціна" збільшення одного з критеріїв і як це позначиться на погіршенні інших. Побудова множини Парето є необхідним при вирішенні багатокритеріальних задач вибору в системах (управління, проектування промислових і транспортних об'єктів, портів тощо).

Відзначимо ще одну важливу особливість альтернатив з множини Парето: кожна з них представляє цілий клас (групу) рішень, що перевершують інші по одному або декільком критеріям. Пояснимо це прикладом. Нехай є навчальна група (множина альтернатив), потрібно вибрати найкращого студента (альтернативу) по ряду критеріїв, наприклад кмітливість, успішність, манера поведінки, зовнішній вигляд, вміння висловлювати свої думки і тощо. Припустимо, що студент x_1 – найкмітливіший, а за іншими критеріями не виділяється. Студенти x_2, x_3, x_4, x_5 мають високі значення інших критеріїв такі, що вони в середньому перевершують x_1 , причому x_2 краще за всіх по успішності, а за іншими критеріями не гірше за інших

студентів. Тоді x_1 обов'язково потрапляє в множину Парето, оскільки він унікальний за першим критерієм, а від групи студентів $x_2 \dots x_5$ в Парето потрапляє один представник – x_2 , хоча інші студенти перевершують x_1 за кількома критеріями.

Після того, як побудовано множину Парето, для визначення найкращого рішення з решти застосовуються в залежності від умов завдання методи першої групи: метод згортки, метод головного критерію і тощо або графічні методи, наприклад метод діаграм.

2.7 Пошук найкращого рішення при впливі навколишнього середовища

Рішення багатокритеріальних задач вибору ще більше ускладнюється, якщо досліджувана система взаємодіє з навколишнім середовищем. У цьому випадку рішення залежить від так званих неконтрольованих параметрів. Наприклад, для вимірювальних систем це можуть впливати величини (температура, тиск тощо), для транспортних – погода, стан дороги тощо. Неконтрольовані зміни стану навколишнього середовища є додатковим джерелом неоднозначності вибору найкращого рішення. Розглянемо дві полярні стратегії вибору, що дозволяють отримати обґрунтовані рішення.

2.7.1 Метод гарантованого результату

Стратегія, заснована на принципі найгіршої реакції навколишнього середовища, отримала назву метод гарантованого результату. Цей метод застосовується, коли середовище поводить себе непередбачувано або вороже (природне середовище). В

цьому випадку визначити найкраще рішення не є можливим, оскільки невідома поведінка середовища, але можна визначити гарантоване рішення, яке справедливо при будь-якому стані середовища. Позначимо a – неконтрольований параметр, що характеризує стан навколишнього середовища (він може бути векторним), $a \in G_a$, де G_a – деяка множина, наприклад інтервал значень. Тоді, і часткові критерії K_j і загальний критерій K залежатимуть від параметру a , тобто $K_j = K_j(x, a)$, $K = K(x, a)$. Принцип найгіршої реакції середовища поширює схему вибору за найгіршим критерієм (максиміну згортку) на випадок впливу навколишнього середовища. Альтернатива вибирається з умови:

$$x^* = \arg \max_{x \in X} \min_{a \in G_a} K(x, a), \quad (2.13)$$

де $K(x, a)$ – загальний критерій, що одержується згортою по вихідним (частковим) критеріям.

Рішення (2.13) називається гарантованим результатом, бо при будь-якому значенні параметру a гарантовано здобуття критерію не менш, ніж $\min_{a \in G_a} K(x, a)$.

2.7.2 Принцип Неша

Часто дії навколишнього середовища є цілеспрямованими, наприклад, це має місце для систем, що включають суб'єктів, при чому кожна з систем прагне досягти своєї мети. Випадок неузгодженості цілей суб'єктів називається конфліктом. При аналізі конфліктів з багатьма суб'єктами одна з важливих проблем – це проблема колективних рішень, або компромісу. Для прийняття рішень в таких системах зберігає своє значення принцип Парето. Ефективні альтернативи, що належать множині

Парето, володіють тією властивістю, що поліпшити значення цільової функції (критерію) будь-якого суб'єкта можна тільки за рахунок інших суб'єктів. Поряд з принципом Парето широко використовується принцип рівноваги, що називається також принципом стійкості, або принципом Неша [10]. Цей принцип дозволяє звужити множину альтернатив, коли мова йде про колективне рішення, що приймається всіма взаємодіючими суб'єктами за домовленістю, при цьому кожен поступається частиною своїх інтересів. Визначимо рівноважне рішення як таке, що приймається всіма суб'єктами одночасно, по домовленості. Нехай є N суб'єктів, кожен з яких може вибирати своє рішення (стратегію) $x^{(l)} \in X^{(l)}$ так, щоб максимізувати свій критерій $K^{(l)}$. Значення критерію при цьому залежить від вибору інших суб'єктів, тобто $K^{(l)} = K^{(l)}(x^{(1)}, \dots, x^{(l)}, \dots, x^{(N)})$. Рішення $x_0 = (x_0^{(1)}, \dots, x_0^{(l)}, \dots, x_0^{(N)})$ називається рівноважним, якщо для будь-якого l виконується умова:

$$K^{(l)}(x_0) = \max_{x^{(l)}} K^{(l)}(x_0^{(1)}, \dots, x_0^{(l-1)}, x^{(l)}, x_0^{(l+1)}, \dots, x_0^{(N)}). \quad (2.14)$$

Рівноважне рішення можна назвати стійким, тому що якщо суб'єкт l відступить від свого рівноважного рішення, тобто обере стратегію $x^{(l)} \neq x_0^{(l)}$, то за умови, що інші суб'єкти збережуть свій вибір, він програє. Принцип Неша якраз і полягає в тому, що найкращі рішення належать множині рівноважних рішень.

2.8 Результати аналізу існуючих методів

Проаналізувавши велику кількість методів розв'язання задачі багатокритеріального вибору та врахувавши те, що робота стосується планування

подорожі, ми маємо можливість зробити наступні висновки щодо подальшого використання цих методів для вирішення поставленої у роботі задачі:

- ми не можемо застосовувати методи першої групи для зменшення множини альтернатив, оскільки нам потрібно запропонувати користувачеві найкращі варіанти рішення, тому принцип Парето нам не підходить для задачі, поставленої у роботі;

- оскільки майже неможливо визначити граничні значення багатьох критеріїв, доцільно буде застосовувати лише методи аддитивної та мультиплікативної згортки.

Найкращим методом для розв'язання поставленої задачі було обрано аддитивну згортку. Мультиплікативна згортка не є найкращим рішенням, оскільки ми матимемо бульові критерії, які матимуть нормалізовані значення 0 та 1, і при значенні критерію 0 результат також буде 0, не зважаючи на інші критерії та їх пріоритети.

Обравши метод ранжування багатокритеріальних альтернатив, ми тепер повинні перевірити наш вибір, розробивши програмну систему, що підтвердить наші твердження.

3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

3.1 UML-модельовання системи

Модельовання розроблюваної системи за допомогою мови UML полягає в побудові діаграм, що допоможуть наочно відобразити функціональність та внутрішню структуру системи. UML – це мова графічного опису для об'єктного модельовання в області розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. При проектуванні системи було розроблено діаграму прецедентів для наочного відображення можливостей користувача у системі.

Діаграма прецедентів (рис.3.1) має одного актора – користувача.

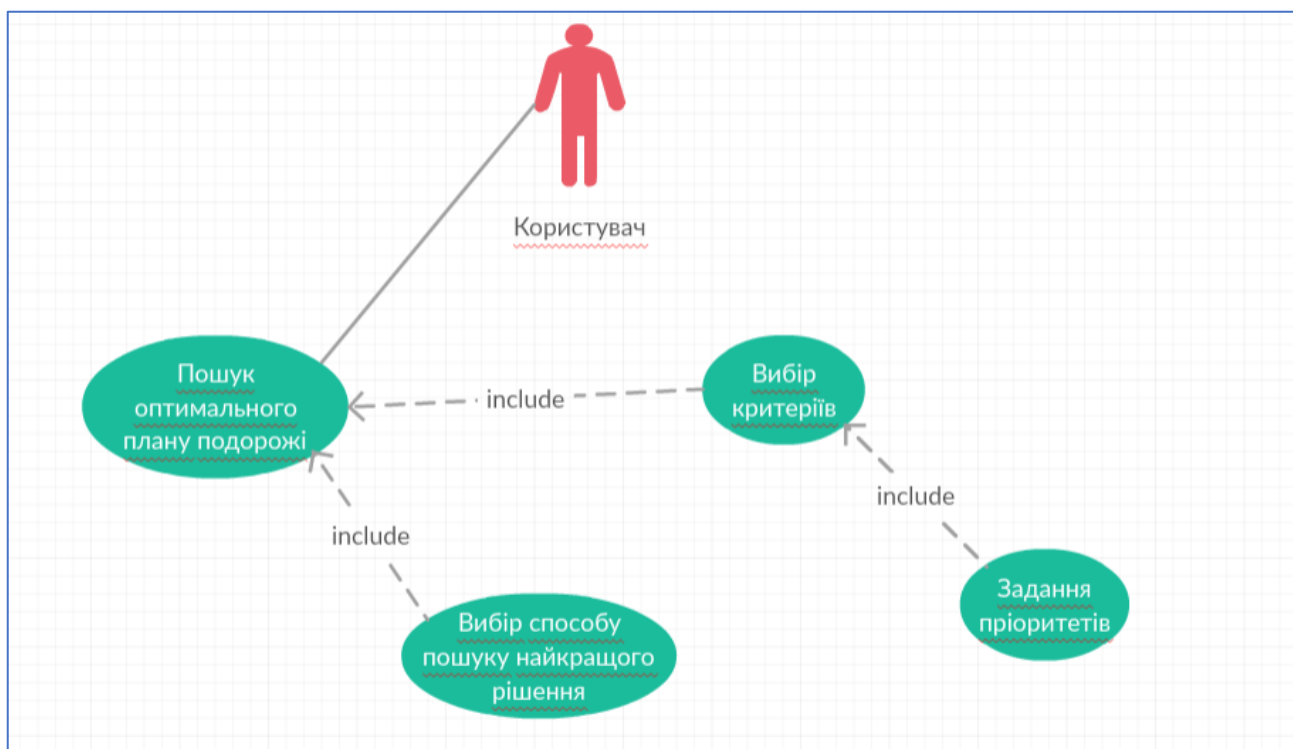


Рисунок 3.1 – Діаграма прецедентів

Користувач – людина, яка користується системою. Він може обрати для себе критерії найкращого плану подорожі разом з пріоритетами. Також користувач може обрати спосіб пошуку найкращого рішення.

На діаграмі послідовностей (рис.3.2) зображено систему в цілому на вищому рівні абстракції та взаємодію основних компонентів – клієнту, серверу та сторонньої системи.

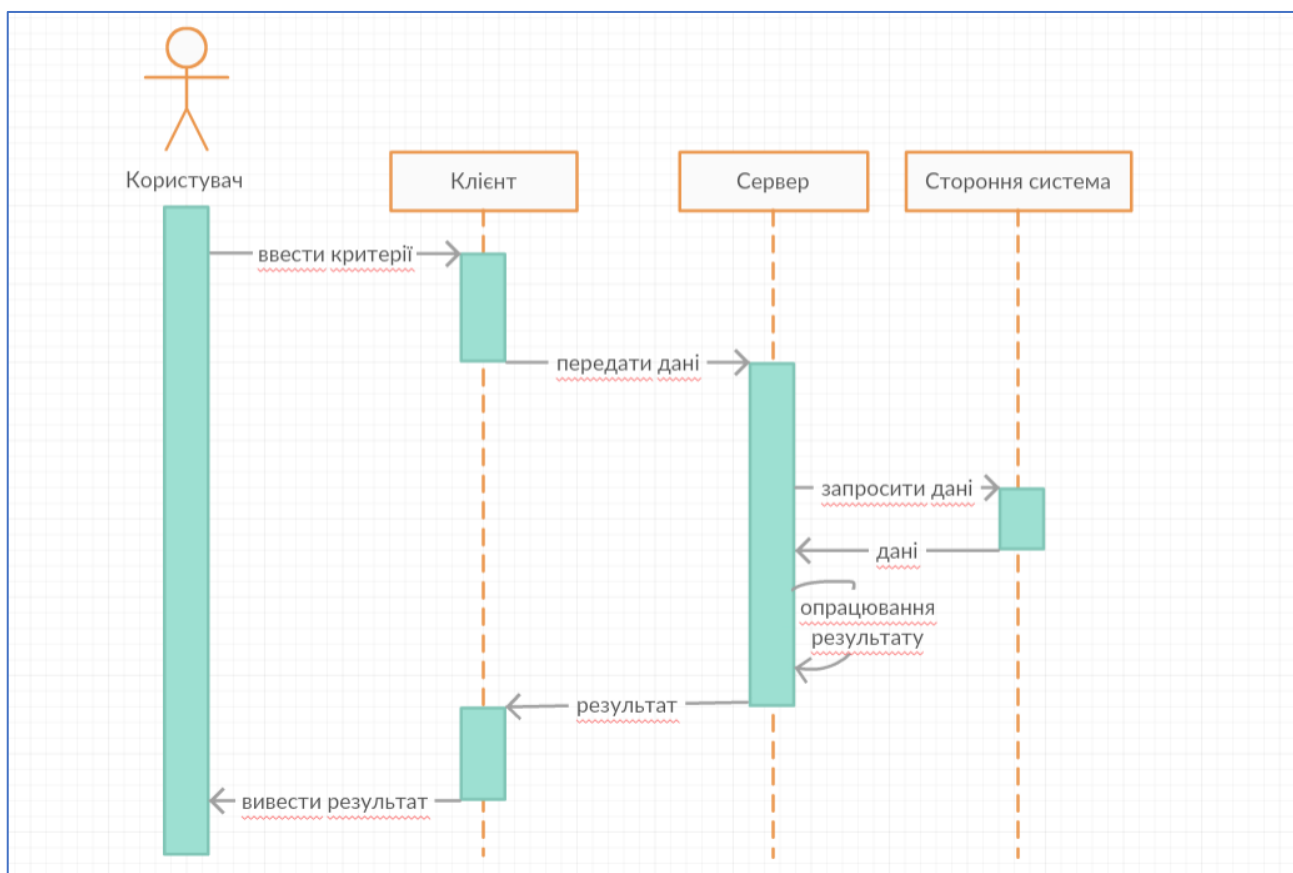


Рисунок 3.2 – Діаграма послідовностей

Користувач контактує з клієнтом та передає через нього дані, які опрацьовує сервер, також запросивши інформацію зі сторонньої системи.

3.2 Розробка моделі системи визначення найкращої альтернативи

Оскільки нам потрібно запропонувати користувачеві упорядковані найкращі рішення, було вирішено не застосовувати принцип Парето для зменшення кількості альтернатив.

Методом знаходження найкращої альтернативи було обрано аддитивну згортку. Для її використання користувач визначає критерії, за якими проводитиметься оцінка альтернатив. Після цього він визначає для них пріоритети (ваги), при чому для рівноважних критеріїв – однакові.

Аби застосовувати метод згортки для порівняння альтернатив, потрібно спочатку зробити нормалізацію критеріїв та їх пріоритетів (сформувавши з них вагові коефіцієнти).

3.2.1 Нормалізація критеріїв

Оцінки за різними критеріями можуть мати різні шкали, наприклад, ціна вимірюється в грн, кількість пересадок в одиницях, відстань готелю від центру в км тощо. Для якісних оцінок використовується бальна шкала, а булеві критерії мають значення $[0|1]$.

Щоб уникати таких непорозумінь, оцінки за критеріями роблять безрозмірними і ставлять так, щоб вони брали значення від 0 до 1.

Нехай у нас є n критеріїв, при цьому для варіанта x оцінки за критеріями наступні:

$$0 \leq f_i(x) \leq 1, i = 1 \dots n. \quad (3.1)$$

Якщо по i -ому критерію x – бездоганий варіант (принаймні, інші варіанти вибору не краще його за даним критерієм), то оцінка дорівнює 1, якщо огидний, то 0, а якщо такий собі – десь між 0 і 1. Виставляти таку оцінку більш чітко допомагає нормування критеріїв.

Один зі способів нормалізації критеріїв – це коли ми знаємо еталонне значення критерію:

$$f = \frac{f_{\text{вимір}}}{f_{\text{max}}}, \quad (3.2)$$

де $f_{\text{вимір}}$ – оцінка критерію для альтернативи;

f_{max} – оцінка еталонного значення критерію.

Але дуже часто буває, коли ми не можемо точно визначити граничні значення критеріїв.

У нас є кілька варіантів. У кращого з точки зору критерію K оцінка за критерієм K становить f_{max} , у гіршого - f_{min} . Тоді нормування для $f_{\text{вимір}}$ виконується так:

$$f = \frac{f_{\text{вимір}} - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}. \quad (3.3)$$

Якщо ж нам потрібна мінімізація, то обчислюємо нормовані оцінки як:

$$f = \frac{f_{\text{max}} - f_{\text{вимір}}}{f_{\text{max}} - f_{\text{min}}}. \quad (3.4)$$

Формули (3.3) та (3.4) зручні тим, що всі $f \in [0; 1]$, при чому $f'_{\text{min}} = 0$, $f'_{\text{max}} = 1$. Таким чином, нормалізований частковий критерій показує, на яку частину всього діапазону змін $[0; 1]$ даний частковий критерій перевершує мінімальне значення.

Отже, нормалізацію проводять різними способами – від застосування більш грубих шкал при вимірюванні оцінок, до обчислення різного роду статистик. Будемо користуватися формулами (3.2 – 3.4). Далі треба визначити вагові коефіцієнти.

3.2.2 Визначення вагових коефіцієнтів

Якщо ви стоїте перед вибором рішення з кількох можливих варіантів, потребується як слід все «зважити». Тому, як вже було зазначено раніше, після визначення критеріїв, потрібно зробити ранжування критеріїв.

Ваговий коефіцієнт – це параметр, який відображає значимість (або «вагу») даного чинника або показника в порівнянні з іншими чинниками, що впливають на досліджуваний об'єкт або явище.

Вагові коефіцієнти показують вплив критерію на результат згортки та визначаються наступною формулою:

$$0 < p_i < 1, i = 1 \dots n; p_1 + \dots + p_n = 1. \quad (3.5)$$

Тепер залишилося з пріоритетів критеріїв отримати вагові коефіцієнти. Скористаємось найпростішим рішенням – простим ранжуванням.

У нас є n критеріїв. Найважливішому присвоюємо n очок, наступному за важливістю $n - 1$ очок, останньому – 1 очко. Отримані оцінки ділимо на $(1 + 2 + \dots + n)$ – так, щоб сума ваг дорівнювала 1.

Нехай кількість пересадок важливіше вартості, а вартість важливіше типу авіалінії. Тоді кількість пересадок – 3 очки, вартість – 2 очки, тип авіалінії – 1 очко. $1 + 2 + 3 = 6$, тому кількість пересадок: $3/6 = 0,5$; вартість: $2/6 = 0,33$; тип авіалінії: $1/6 = 0,17$.

Виникає питання: а якщо є критерії з рівними пріоритетами, наприклад, кількість пересадок важливіше за все інше, але вартість квитка та тип авіалінії мають рівні пріоритети. Тоді маємо наступне: на вартість квитка та тип авіалінії в сумі має припасти 3 очки ($1 + 2$), але оскільки ці критерії рівноцінні, перерозподіляємо ці 3 очки порівну: по 1,5 кожному. На кількість пересадок як і раніше "виділено" 3 очки. У підсумку отримуємо ваги $3/6 = 0,5$; $1,5/6 = 0,25$; $1,5/6 = 0,25$.

Приклад складніше: користувач визнав, що однаково важливі і важливіше за все інше кількість пересадок і вартість квитка, менш важливі за них тип авіалінії та присутність WI-FI, при чому останні два критерії мають однаковий пріоритет. На перші два критерії припадає $4 + 3 = 7$ очок, ділимо їх порівну і отримуємо 3,5 очки на кожен. На останні два: $2 + 1 = 3$ очки, ділимо порівну і отримуємо по 1,5 очки. Нормуємо розподілом на $1 + 2 + 3 + 4 = 10$ і отримуємо наступні ваги: 0,35; 0,35; 0,15; 0,15.

Отже, ми маємо нормалізовані критерії з ваговими коефіцієнтами, тому можемо переходити до наступного етапу – застосування методу згортки.

3.2.3 Використання методів згортки

Як вже було зазначено, багатокритеріальну задачу прийняття рішень на множині альтернатив можна попередньо звести до однокритеріальної, ввівши якийсь узагальнений критерій $w(x)$, як функцію від попередніх часткових критеріїв. Узагальнений критерій $w(x)$ в літературі також називають функцією корисності. Цей процес зведення багатокритеріальної задачі до однокритеріальної називається згорткою.

Оскільки ми визначилися зі способами нормування критеріїв та визначення вагових коефіцієнтів, можемо застосовувати метод згортки, а саме аддитивної згортки, що була обрана для розв'язання задачі.

Маючи формули (3.1) та (3.5), визначимо оператор згортки критеріїв для альтернативи x :

$$w(x) = g(f_1(x), \dots, f_n(x), p_1, \dots, p_n) = \sum_{j=1}^n p_j f_j(x). \quad (3.6)$$

Тепер застосуємо (3.6) для визначення найкращої альтернативи. Припустимо, ми маємо таку таблицю критеріїв:

Таблиця 1 – Вхідна множина альтернатив з критеріями

Альтернатива / Критерій	Кількість пересадок, (одиниці)	Вартість, (грн)	WI-FI, (+ або -)	Тип авіалінії, (бальна шкала)
A1	1	100	+	Lufthansa (2)
A2	0	130	-	Swiss (1)
A3	0	80	-	KLM (3)
A4	2	50	-	KLM (3)
A5	0	110	+	Swiss (1)
A6	1	60	+	Lufthansa (2)

За даними з таблиці 1 визначимо найкращу альтернативу за формулою:

$$x^* = \arg \max_{x \in X} w(x). \quad (3.7)$$

Спочатку використаємо аддитивну згортку з нормалізованими критеріями без урахування вагових коефіцієнтів. Для цього використаємо (3.4) для мінімізації (кількість пересадок, вартість тощо) та (3.3) для максимізації (рейтинг, WI-FI тощо).

Результат наведемо у таблиці:

Таблиця 2 – Аdditивна згортка з нормалізованими критеріями

Альтернатива / Критерій	Кількість пересадок	Вартість	WI-FI	Тип авіалінії	Результат згортки з нормалізованими критеріями
A1	0,5	0,375	1	0,5	2,375
A2	1	0	0	1	2
A3	1	0,625	0	0	1,625
A4	0	1	0	0	1
A5	1	0,25	1	1	3,25
A6	0,5	0,875	1	0,5	2,875

Як бачимо, оптимальною стратегією є A5. Аdditивна згортка з нормалізованими критеріями дозволяє працювати з якісними та кількісними критеріями, які мають, як в нашому випадку, різні одиниці вимірювань.

Аdditивна згортка особливо гарна, коли зменшення оцінки за якимось одним критерієм компенсується збільшенням оцінки по якомусь іншому критерію (або кільком критеріям). Неважко помітити: навіть якщо по якимось критеріям оцінка нульова, єдина оцінка може вийти цілком пристойною, якщо за іншими критеріями все не так погано.

Тепер розглянемо аdditивну згортку з ваговими коефіцієнтами. Зрозуміло, що в кожній окремій ситуації часткові критерії по-різному впливають на загальний суперкритерій. Для того, щоб позначити цей вплив їм надають в загальній формулі різні ваги. Це можна зробити за допомогою вагових коефіцієнтів.

Отже, припустимо, користувач вказав пріоритети: найважливіше для нього – це вартість, потім кількість пересадок, а тип авіалінії та наявність WI-FI – однаковопріоритетні. Тоді на вартість виділяємо 4 очки, кількість пересадок – 3 очки, інші два критерії – $1 + 2 = 3$ очки, тобто на кожен з них виходить по 1,5 очки. Загальна кількість очок виходить $1 + 2 + 3 + 4 = 10$.

Тому, привівши вагові коефіцієнти до (3.5), в нас виходить така функція оцінки альтернативи:

$$w(x) = 0,3f_1 + 0,4f_2 + 0,15f_3 + 0,15f_4 \quad (3.8)$$

Застосуємо формулу (3.8) до наших альтернатив та отримаємо таку таблицю:

Таблиця 3 – Лінійна аддитивна згортка з ваговими коефіцієнтами

Альтернатива / Критерій	Кількість пересадок	Вартість	WI-FI	Тип авіалінії	Результат згортки з ваговими коефіцієнтами
A1	0,5	0,375	1	0,5	0,525
A2	1	0	0	1	0,45
A3	1	0,625	0	0	0,55
A4	0	1	0	0	0,4
A5	1	0,25	1	1	0,7
A6	0,5	0,875	1	0,5	0,725
Вагові коефіцієнти	0,3	0,4	0,15	0,15	

Як бачимо, при незмінній умові завдання трохи більш оптимальною вийшла стратегія A6, хоча в попередньому випадку без урахування вагових коефіцієнтів переважила стратегія A5. Та й стратегія A3, що була на передостанньому місці, вийшла не такою вже й несприятливою. Ось чому так важливі пріоритети критеріїв.

Отже, згідно з результатами згортки, наведених у таблиці 3, та формулою (3.7), альтернативи планів подорожі пропонуються користувачеві у наступному порядку: A6, A5, A3, A1, A2, A4.

3.2.4 Аналіз результатів згортки

Ми проаналізували аддитивну згортку та визначили алгоритм її застосування, який включає наступну послідовність дій:

- застосовуючи формули нормалізації критеріїв (3.3) та (3.4), потрібно нормалізувати оцінки альтернатив таким чином, щоб вони відповідали формулі (3.1);
- визначити вагові коефіцієнти критеріїв, трансформуючи ранжовані пріоритети від користувача, таким чином, щоб вони відповідали формулі (3.5);
- застосувати оператор згортки (3.6) та впорядкувати альтернативи за зменшенням узагальненого критерію.

Отже, застосувавши цю математичну модель, далі ми повинні розробити програмну систему для перевірки гіпотези щодо можливого застосування ранжованого підходу до багатокритеріального пошуку та порівняти результати роботи програми з тестовими даними, наведеними у таблиці 1, з вихідним порядком результатів з таблиці 3, отриманих під час теоретичного застосування згортки.

3.3 Розробка архітектури застосунку

Для реалізації системи було вирішено використовувати клієнт-серверну архітектуру, адже вона найбільше підходить для виконання поставленої задачі.

Клієнт-серверна архітектура (див. рис.3.3) набула своєї популярності завдяки динамічному розвитку мережі Інтернет та зосередженню значної частини інформації в базах даних на серверах.

Клієнт-серверну архітектуру можна означити, як концепцію інформаційної мережі в якій основна частина її ресурсів зосереджена в серверах, що обслуговують своїх клієнтів. Така архітектура визначає наступні типи компонентів:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

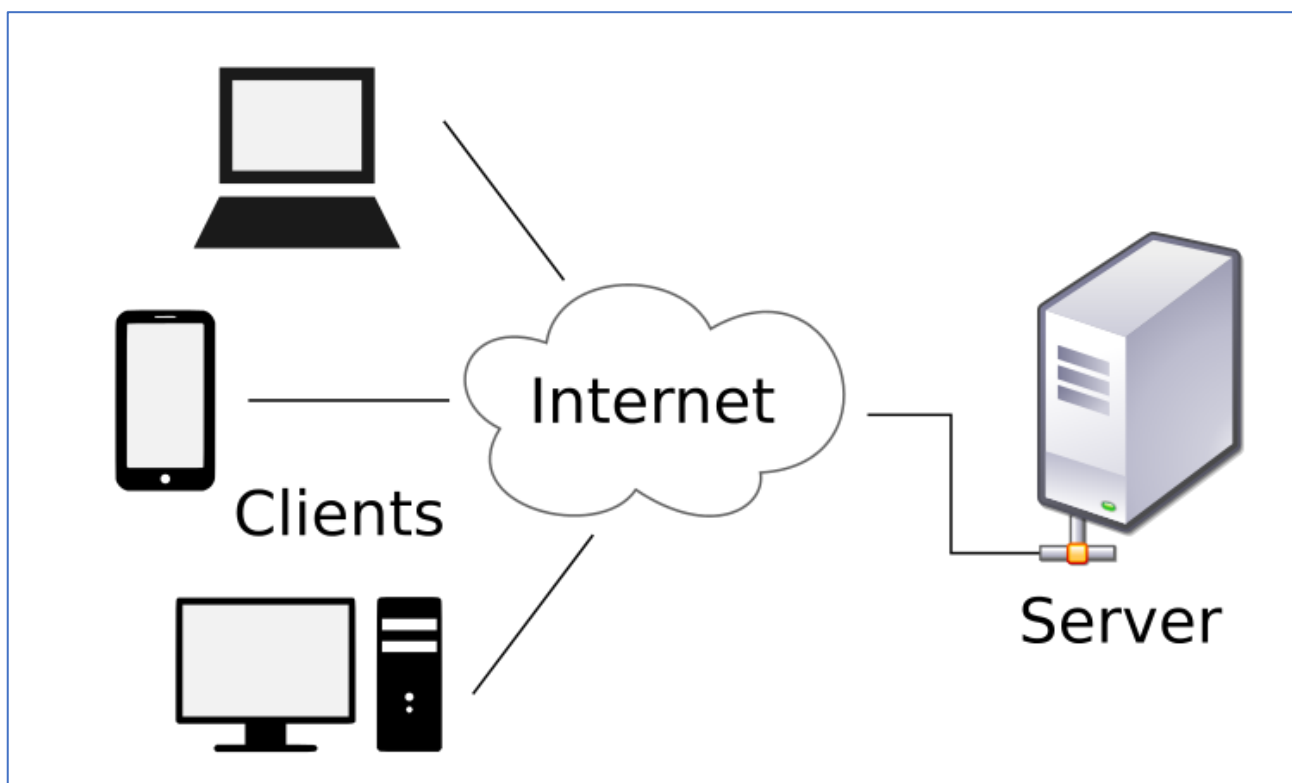


Рисунок 3.3 – Клієнт-серверна архітектура

Правила взаємодії між клієнтом і сервером називаються протоколом обміну (протоколом взаємодії).

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дволанкова клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка застосунку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони тощо.

Триланкова клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка застосунку. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів застосунків, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Дволанкова архітектура простіша, так як всі запити обслуговуються одним сервером, але саме через це вона менш надійна і висуває підвищені вимоги до продуктивності сервера.

Триланкова архітектура складніша, але завдяки тому, що функції розподілені між серверами другого і третього рівня, ця архітектура проявляє такі властивості:

- високий ступінь гнучкості і масштабованості;
- високу безпеку (тому що захист можна визначити для кожного сервісу або рівня);
- високу продуктивність (тому що завдання розподілені між серверами).

Прикладом клієнт-серверної взаємодії є сервіс WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, що розмічені за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є

динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Основна ідея архітектури «клієнт-сервер» полягає в поділі мережевого додатку на кілька компонентів, кожен з яких реалізує специфічний набір сервісів. Компоненти такого додатку можуть виконуватися на різних комп'ютерах, виконуючи серверні і/або клієнтські функції. Це дозволяє підвищити надійність, безпеку і продуктивність мережевих додатків і мережі в цілому.

Ця архітектура є одним з архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних додатків і передбачає взаємодію та обмін даними між ними.

На рисунку 3.4 наведене схематичне зображення архітектури системи.

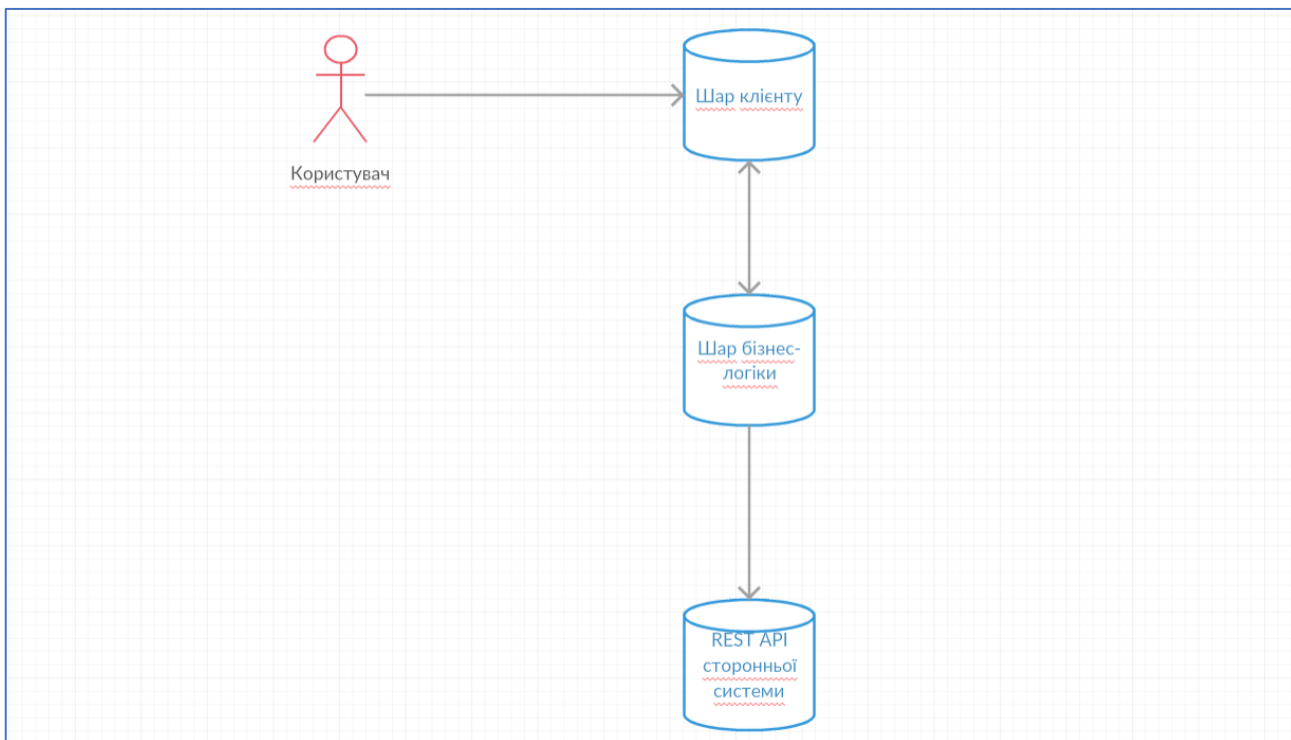


Рисунок 3.4 – Архітектура системи

Архітектура нашої програмної системи має наступні рівні:

- інтерфейс користувача забезпечує шар представлення (Presentation Layer). Завдяки використанню графічного інтерфейсу, користувач безпосередньо взаємодіє зі всією системою;

- шар бізнес-логіки (Business Layer) є посередником між графічним інтерфейсом та сторонньою системою. Він містить усі необхідні функції, які необхідні системі для функціонування. Шар бізнес-логіки має відповідати патернам та принципам проектування програмного забезпечення. Оскільки шари є незалежними одне від одного, шар бізнес-логіки може бути використаний як мобільним, так і веб-додатком, завдяки публічному API;

- стороння система, на яку йде запит від шару бізнес-логіки. Це може бути одна або декілька систем, проаналізованих у першому розділі. Усі вони мають публічний REST API, завдяки якому наш додаток спілкуватиметься з ними.

Взаємодія користувача з програмною системою йде наступним чином: користувач завдяки клієнту працює з шаром представлення, який відображає елементи графічного інтерфейсу. Шар представлення виконує запити до шару бізнес-логіки, завдяки API, який є на сервері. Сервер відсилає запит на сторонню систему і чекає на відповідь, після чого відповідь оброблюється бізнес-логікою, і результат відсилається до клієнту. При цьому сервіси і клієнти є незалежними один від одного, немає жорсткої прив'язки клієнтів до сервісів.

3.4 Опис програмних рішень

Для підтвердження гіпотез про можливість використання багатокритеріального вибору, необхідно було розробити програмну систему. Під час планування та моделювання програмного продукту було визначено, що програмна система має

складатися з двох частин: сервера, який буде отримувати та обробляти дані, та клієнта, що підключатиметься до нього за допомогою публічного REST API.

3.4.1 Вибір засобів розробки

Для розробки серверу використовується технологія створення Web-застосунків мовою Java з використанням SpringBoot фреймворку.

Java – об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році, як основний компонент платформи Java. Java є платформо-незалежною мовою, тому це досить поширена та популярна мова програмування.

Для розробки використовується середовище IntelliJ IDEA Community Edition. IntelliJ IDEA являє собою написану на Java середу розробки загального призначення, архітектура якої забезпечує можливості для вирішення різних завдань, таких як інтеграція різних інструментів і мов програмування. Механізми подібної інтеграції дозволяють використовувати IntelliJ IDEA для побудови розвинених середовищ розробки, звільняють від рутини написання базових засобів на користь створення складних, спеціалізованих функцій. Тим самим не тільки вирішується проблема підтримки багатофункціональних і багатомовних середовищ розробки, але і закладається база для спрощення переходу від одного типу середовища до іншого в процесі їх еволюції.

Для виклику методів сервера використовуються REST контролери з фреймворку Spring, що дозволяє отримувати та відправляти різноформатні дані по протоколу HTTP.

На сьогоднішній день прийнято використовувати REST (Representational State Transfer) – метод взаємодії компонентів розподіленого додатка в мережі Інтернет, при

якому виклик віддаленої процедури являє собою звичайний HTTP-запит, а необхідні дані передаються як параметри запиту.

У свою чергу HTTP – протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від Hyper Text Transfer Protocol, протокол передачі гіпер-текстових документів.

HTTP – протокол прикладного рівня, схожими на нього є FTP і SMTP. Обмін повідомленнями йде за звичайною схемою «запит-відповідь». Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну від багатьох інших протоколів, HTTP не зберігає свого стану. Це означає відсутність збереження проміжного стану між парами «запит-відповідь».

Для кешування даних, що надходять зі сторонньої системи, була використана MongoDB – крос-платформенна документ-орієнтована програма бази даних. Класифікована як програма бази даних NoSQL, MongoDB використовує JSON-подібні документи. MongoDB розроблена компанією MongoDB Inc. і ліцензована під ліцензією Server Side Public License (SSPL).

До основних плюсів MongoDB можна віднести реплікацію, файлову систему збереження, балансування навантаження та агрегацію.

Веб-сервіс створений на базі контейнеру сервлетів Spring і використовує повний спектр інструментів Spring Boot Framework.

Платформа Spring – популярна платформа додатків з відкритим кодом, призначена для спрощення розробки для J2EE. Вона складається з контейнера, платформи управління елементами і набору служб для веб-інтерфейсів користувача, транзакцій і збереження стану.

Spring Boot – розширювана платформа для створення веб-додатків, що входить до складу платформи Spring. Spring Framework складається з декількох модулів, які надають широкий спектр послуг:

- інверсія управління: конфігурація компонентів додатків і управління життєвим циклом об'єктів Java здійснюється переважно через Інверсію управління;

- аспект-орієнтоване програмування: дозволяє реалізувати наскрізні процедури;
- доступ до даних: робота з реляційною системою управління базами даних на платформі Java з використанням JDBC і об'єктно-реляційних відображень та інструментів з NoSQL баз даних;
- управління транзакціями: об'єднує кілька API, управління транзакціями і координує операції для Java-об'єктів;
- Model-View-Controller: HTTP сервлети, що забезпечує створення веб-додатків і веб-служб RESTful;
- автентифікація і авторизація: настраюються процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик за допомогою Spring Security суб-проекту;
- тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів.

Для розробки Front-end веб-клієнту використовується Angular, а для розробки серверу використовується фреймворк SpringMVC.

В якості системи керування версіями обрано Git. Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів.

3.4.2 Опис програмної системи

Веб-клієнт має лише одну сторінку, де користувач спочатку обирає дати та маршрут та визначає критерії, за якими буде зроблено пошук (див. рис.3.5).

Рисунок 3.5 – Користувач обирає критерії

Для якісного критерію користувачеві пропонується проранжувати можливі значення (рис.3.6).

Рисунок 3.6 – Користувач ранжирує можливі значення

Після цього він задає їм пріоритети (див. рис.3.7). У разі якісного критерію з'явиться кнопка для оцінки можливих значень.

The screenshot shows a web application for ranking criteria. At the top, there is a list of criteria: 'Transitions count', 'Airline type', 'Price', 'Class', 'Wi-Fi in hotel', and 'Hotel rating'. Below this, a section titled 'Set criteria ranks (from 1 for lowest)' contains input fields for 'Transitions count' (value 2), 'Airline type' (value 1), 'Price' (value 3), and 'Wi-Fi in hotel' (value 1). A 'Mark possible values' button is located next to the 'Airline type' field. At the bottom of the form is a 'Search plan' button.

Рисунок 3.7 – Користувач ранжирує критерії

Після натискання на кнопку «Шукати план», дані, в тому числі критерії з їх пріоритетами, відправляються на сервер.

Далі починає здійснювати обробку сервер. Спочатку він отримує дані зі сторонньої системи завдяки REST API. Для того щоб почати їх обробку, потрібно нормалізувати критерії. Наведемо програмну реалізацію функції нормалізації критеріїв:

```
public double normalize(double x) {
    return ((x - dataLow)
        / (dataHigh - dataLow))
        * (normalizedHigh - normalizedLow) + normalizedLow;
}
```

Після того, як було зроблено нормалізацію, система вираховує загальну оцінку альтернативи. Наведемо програмну реалізацію функції отримання загальної оцінки (застосування методу згортки):

```
public double result(List<Pair<Double, Double>> criteriaList) {
```

```

return criteriaList.stream()
    .mapToDouble(pair -> pair.getKey() * pair.getValue())
    .sum();
}

```

Далі альтернативи упорядковуються за результатами згортки і повертаються назад до клієнта.

На рисунку 3.8 відображено результат роботи програмної системи:

Rome - London 20.06.2019 - 27.06.2019				
Transitions count: 1	Price: 60 uah	Wi-Fi in hotel: +	Airline type: Lufthansa	View detailed info

Rome - London 20.06.2019 - 27.06.2019				
Transitions count: 0	Price: 110 uah	Wi-Fi in hotel: +	Airline type: Swiss	View detailed info

Rome - London 20.06.2019 - 27.06.2019				
Transitions count: 0	Price: 80 uah	Wi-Fi in hotel: -	Airline type: KLM	View detailed info

Rome - London 20.06.2019 - 27.06.2019				
Transitions count: 1	Price: 100 uah	Wi-Fi in hotel: +	Airline type: Lufthansa	View detailed info

Rome - London 20.06.2019 - 27.06.2019				
Transitions count: 0	Price: 130 uah	Wi-Fi in hotel: -	Airline type: Swiss	View detailed info

Rome - London 20.06.2019 - 27.06.2019				
Transitions count: 2	Price: 50 uah	Wi-Fi in hotel: -	Airline type: KLM	View detailed info

Рисунок 3.8 – Результат роботи програмної системи

Як ми бачимо, результат роботи програмної системи повністю відображає результат наших досліджень. Альтернативи відображаються в порядку спадання привабливості для користувача згідно з зазначеними критеріями та їх пріоритетами.

Отже, результатом роботи програмної системи ми підтвердили гіпотезу щодо можливого застосування ранжованого підходу до багатокритеріального пошуку.

ВИСНОВКИ

В результаті написання атестаційної роботи були проаналізовані існуючі аналоги, що є найпопулярнішими у світі подорожей та стосуються авіаперевезень, бронювання готелів, пошуку закладів для харчування тощо. Були досліджені їх переваги у тій чи іншій області та загальні недоліки, проаналізовані їх фінансові успіхи на міжнародному ринку. Згідно з цим було сформовано мету атестаційної роботи.

Також були проаналізовані існуючі методи розв'язання задачі багатокритеріального вибору, розглянуті їх переваги та недоліки, та визначено, які саме методи підходять для розв'язання поставленої у роботі задачі, а також зазначено чому.

Було визначено, що для розв'язання задачі багатокритеріального вибору використовуються дві групи методів. У методах першої групи прагнуть скоротити число критеріїв, для чого вводять додаткові припущення, що відносяться до процедури ранжування критеріїв і порівняння альтернатив. У методах другої групи прагнуть скоротити число альтернатив у вихідній множині, виключивши свідомо погані.

Під час детального аналізу методів розв'язання задачі багатокритеріального вибору було прийнято рішення не застосовувати методи другої групи для скорочення кількості альтернатив, оскільки поставлена у даній атестаційній роботі задача впорядкування альтернатив відноситься до планування подорожі, і саме тому ми маємо запропонувати користувачеві результати в порядку зменшення їх привабливості.

З методів першої групи для розв'язання задачі було обрано метод аддитивної згортки. Вона базується на введенні інтегрального критерію і подальшому зведенні

вихідної багатокритеріальної задачі до скалярної і вона також дуже зручна у використанні.

Для застосування аддитивної згортки було сформовано алгоритм, що передбачає попередню нормалізацію критеріїв, зважаючи на їх різні шкали оцінювання, та визначення вагових коефіцієнтів, зважаючи на ранжовані пріоритети. Для нормалізації були вибрані можливі критерії оцінювання альтернатив та шкали їх оцінювання.

Також було спроектовано та розроблено програмну систему для перевірки гіпотези щодо можливого застосування ранжованого підходу до багатокритеріального пошуку.

.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Волошин А., Маляр Н., Швалагин О. Нечіткий алгоритм полідового аналізу варіантів.//International Book Series “Information Science & Computing”. 2009. Vol 3. № 15. P. 189 – 195.
2. Ногін В. Д. Прийняття рішень в багатокритеріальному середовищі: кількісний підхід. – Москва: ФІЗМАТЛІТ, 2012. – 144 с. – ISBN 5-9221-0274-5.
3. Ротштейн А. П., Штовба С. Д. Нечіткий багатокритеріальний аналіз варіантів із застосуванням парних порівнянь.//Звістки РАН. Теорія та системи управління. 2011. №3. С. 150 – 154.
4. Борисов А. Н., Крмберг О. А., Федоров І. П. Прийняття рішень, засновуючись на нечітких моделях: приклади використання. – Рига: Зінатне, 2010. – 184 с.
5. Андрейчикова О. А. Прийняття рішень в умовах взаємної залежності критеріїв та альтернатив складних технічних систем.//Інформаційні технології. 2011. №11. С. 14 – 19.
6. Voloshyn A. Decision-Making Support Systems as Personal Intellectual Device of a Decision-Maker.//International Journal “Information: Technologies & Knowledge”. 2017. Vol 1. № 2. P. 159 – 162.
7. Voloshyn O., Antosiak P. Procedures of Sequential Analyses and Sifting of Variants for the Linear Ordering Problem.//International Book Series “Information Science & Computing”. 2009. № 15. P. 149 – 154.
8. Voloshyn O., Berezovskiy K. Developing collective teaching computer software for the course “Decision theory”.//International Journal “Information Technologies & Knowledge”. 2017. Vol 1. P. 33 – 36.
9. Huang J. J., Ong C. S., Tzeng G. H. Optimal fuzzy multi-criteria expansion of competence sets using multi-objectives evolutionary algorithms.//Expert Systems with Applications. 2016. № 30. P. 739 – 745.

10. Hillier F. S. Evaluation and decision models: A critical perspective. – Boston: Kluwer Academic Publishers, 2011. – 211 p.
11. Delgado M., Verdegay J. L., Vila M. A. Linguistic decision-making models.//International Journal of Intelligent System. 2012. № 7. P. 479 – 492.
12. Chu M. T., Shyu J., Tzeng G. H. Comparison among three analytical methods for knowledge communities group-decision analysis.//Expert Systems with Applications. 2017. № 33. P. 1011 – 1024.
13. Chou T. Y., Chou S. T., Tzeng G. H. Evaluating IT/IS investments: A fuzzy multi-criteria decision model approach.//European Journal of Operational Research. 2016. № 173. P. 1026 – 1046.
14. Belton V., Stewart T. J. Multiple criteria decision analysis: An integrated approach. – Boston: Kluwer Academic Publishers, 2012. – 167 p.
15. Armstrong R. D., Cook W. D., Seiford L. M. Priority ranking and consensus formation.//Management Science. 2012. № 28. P. 638 – 645.