

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

**на тему «Інформаційна технологія попарного порівняння з передбаченням
відповіді»**

Виконав: студент 2 курсу,
групи 2КН-19 м
спеціальності 122 «Комп'ютерні науки»
Хохлов А. Д.

Керівник к.т.н., доц. Колодний В. В.

Рецензент к.т.н., доц. Романюк О. В.

Вінниця
2020

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

ЗАТВЕРДЖЕНО
Завідувач кафедри КН
проф., д.т.н. Яровий А. А.
«___» _____ 2020 р.

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра наук зі спеціальності: 122 «Комп'ютерні науки»
(шифр – назва спеціальності)

08-22.МКР.024.19.000.ПЗ

Магістранта групи 2КН-19м Хохлова Андрія Дмитровича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія попарного порівняння з передбаченням відповіді»

Вхідні дані: ранжування альтернатив, об'єктно-орієнтована мова програмування, наявність графічного інтерфейсу, операційна система Windows.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Алгоритм роботи модуля попарного порівняння; Декомпозиція модуля попарного порівняння за методологією SADT; Стартове вікно програмного модуля.

2. Текстова (пояснювальна записка): Вступ, аналіз предметної області, розробка модуля попарного порівняння, тестування роботи програми, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного рівня розвитку інформаційних технологій попарних порівнянь. Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Розробка методу та інформаційної технології попарного порівняння з передбаченням відповіді			Метод, інформаційна технологія, розділ 2
3	Програмна реалізація розробленої інформаційної технології, тестування та оцінка параметрів			Програмне забезпечення, розділ 3
4	Підготовка економічної частини			розділ 4
5	Апробація та/або впровадження результатів дослідження			тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)

“ ____ ” _____ 20__ р. _____ Колодний В.В.
ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доц. доц. каф. ЕПВМ
(підпис) наук. ступінь, вчене звання (посада)

_____ М. В. Бальзан
ініціали та прізвище

“ ____ ” _____ 20__ р.

Дата попереднього захисту роботи “ ____ ” _____ 20__ р.

Рецензент _____ канд. техн. наук, доц. кафедри ПЗ
(підпис) наук. ступінь, вчене звання (посада)

_____ О. В. Романюк
ініціали та прізвище

Завдання видав
науковий керівник

_____ канд. техн. наук, доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)

_____ Колодний В.В.
ініціали та прізвище

“ ____ ” _____ 20__ р.

Завдання отримав магістрант

_____ А.Д. Хохлов
(підпис) ініціали та прізвище

“ ____ ” _____ 20__ р.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РАНЖУВАННЯ АЛЬТЕРНАТИВ	10
1.1 Методи експертних оцінок	10
1.2 Опис теорії прийняття рішень	10
1.3 Огляд відомих методів ранжування	14
1.4 Аналіз існуючих програмних засобів попарного порівняння альтернатив.....	17
1.5 Постановка задачі попарного порівняння альтернатив.....	19
1.6 Висновки.....	20
2 ПРОЕКТУВАННЯ МОДУЛЮ ВІЗУАЛЬНОГО ПОПАРНОГО ПОРІВНЯННЯ З ПЕРЕДБАЧЕННЯМ ВІДПОВІДІ	22
2.1 Структурна організація модулю візуального попарного порівняння з передбаченням відповіді	22
2.1.1 Функціональне моделювання системи.....	22
2.1.2 Проектування архітектури системи.....	27
2.1.3 Проектування системи передбачення відповіді.....	29
2.2 Обґрунтування доцільності застосування попарних порівнянь для ранжування альтернатив	31
2.3 Висновки.....	35
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЮ ВІЗУАЛЬНОГО ПОПАРНОГО ПОРІВНЯННЯ	36
3.1 Обґрунтування вибору технологій для розробки.....	36

3.2 Основні оператори мови програмування javascript	38
3.3 Обґрунтування вибору середовища розробки	39
3.4 Тестовий приклад роботи програми і аналіз результатів	42
3.5 Висновки	46
4 ЕКОНОМІЧНА ЧАСТИНА	48
4.1 Оцінювання комерційного потенціалу розробки	48
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи	49
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки	55
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності	56
4.5 Висновок	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	559
ДОДАТОК А	660
ДОДАТОК Б	61
ДОДАТОК В	79

АНОТАЦІЯ

Метою магістерської роботи є візуальна адаптація алгоритму попарних порівнянь шляхом створення модуля візуального попарного порівняння з можливістю передбачення відповіді. В роботі були проаналізовані алгоритми ранжувань. Було програмно реалізовано алгоритм попарного порівняння з використанням візуального інтерфейсу.

Ключові слова: попарні порівняння, ранжування альтернатив, візуальне порівняння альтернатив.

ABSTRACT

The purpose of the work of master`s work is the visual adaptation of the algorithm of pairwise comparisons by creating a visual parity comparison module with the ability to predict the answer. In this work, algorithms of ranking were analyzed. It was programmed to implement a pairwise comparison algorithm using the visual interface.

Keywords: pairwise comparison, ranking of alternatives, visual comparison of alternatives.

ВСТУП

Актуальність теми дослідження. На сучасному етапі оцінювання застосовують у найрізноманітніших галузях економіки, науки та освітньо-культурної сфери. Особливе місце в оцінюванні та підтримці прийняття рішень займають ординальні оцінки, або ранжування. Ординальне оцінювання має місце у випадках, коли важко, або неможливо визначити точні абсолютні або відносні значення характеристик об'єктів (альтернатив). Ординальною оцінкою (рангом) об'єкта за певним критерієм вважається його номер у заданій множині об'єктів, розташованих у порядку спадання або зростання ступеня виразності даного критерію.

Із стрімким зростанням інформаційних технологій, розробки нових експертних методів для різних прикладних задач користуються великою популярністю, і відповідно, вимагають зручних для широкого кола користувачів рішень.

Прийняття рішення в реальних системах вимагає, як правило, оцінки цілої низки критеріїв. Особливу увагу приділяють на розробку математичного апарату, який би давав можливість, тою чи іншою мірою, зменшити суб'єктивний фактор.

Публікації. Під час виконання даної роботи були опубліковані тези на науково-технічній конференції ВНТУ [1], а також відправлена заявка на отримання авторського права на програмний засіб [2].

Мета та завдання дослідження. Метою даної магістерської роботи є розширення функціональних можливостей системи ранжування альтернатив шляхом створення модуля візуального попарного порівняння з передбаченням відповіді.

Об'єкт дослідження. Об'єктом дослідження є процес прийняття рішень при ранжуванні альтернатив для попарного порівняння.

Предмет дослідження: програмні засоби ранжування альтернатив для попарного порівняння.

Наукова новизна одержаних результатів полягає в наступному:

- Використано збір статистичних даних відповідей користувачів для передбачення відповідей при подібних вподобаннях;
- Розроблено динамічну архітектуру системи, що дозволяє використання попарних порівнянь як SaaS (Програма як послуга) додатку;
- Удосконалено системний інтерфейс програми у порівнянні з конкурентами завдяки використанню привабливого і зручного для користувача інтерфейсу;

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено новий спосіб порівняння сутностей (наприклад товарів) у системі інтернет.
2. Розроблено алгоритм збору статистичних даних при попарному порівнянні.
3. Розроблено алгоритм використання статистичних даних для передбачення відповіді.
4. Розроблено програмний засіб для попарного порівняння, який легко додати у існуючий проект.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими, та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РАНЖУВАННЯ АЛЬТЕРНАТИВ

1.1 Методи експертних оцінок

Найчастіше необхідно вибрати серед безлічі альтернатив, при цьому кожна володіє різними перевагами. І як же вибрати кращу, маючи думку десятків, а то і сотень експертів?

Як обчислення рейтингу комп'ютерної гри, заснованого на оцінках критиками графіки, геймплея і сюжету, так і колективний вибір пріоритетного завдання перед появою замовника, відноситься до методів експертних оцінок.

Методи експертних оцінок є частиною великої області теорії прийняття рішень, а саме експертне оцінювання - процедура отримання оцінки проблеми на основі думки фахівців (експертів) з метою подальшого прийняття рішення (вибору).

У випадках надзвичайної складності проблеми, її новизни, недостатності наявної інформації, неможливості математичної формалізації процесу вирішення доводиться звертатися до рекомендацій компетентних фахівців, прекрасно знають проблему, - до експертів. Їх рішення задачі, аргументація, формування кількісних оцінок, обробка останніх формальними методами дістали назву методу експертних оцінок.

Існує дві групи експертних оцінок:

Індивідуальні оцінки засновані на використанні думки окремих експертів, незалежних один від одного.

Колективні оцінки засновані на використанні колективної думки експертів.

1.2 Опис теорії прийняття рішень

Теорія прийняття рішень – область дослідження, в якій використовуються поняття і методи математики, статистики, економіки, менеджменту і психології, яка вивчає закономірності вибору людьми шляхів вирішення різного роду завдань, а також досліджує способи пошуку найбільш вигідних з можливих рішень. [3].

У наш час теорію прийняття рішень (ТПР) застосовують переважно для аналізу тих проблем, які можна відносно легко й однозначно формалізувати, а результати досліджень – адекватно інтерпретувати.

Задача прийняття рішень (ЗПР) може бути сформульована в термінах мети, способів її досягнення (альтернатив) та отриманих результатів (рис. 1.1).

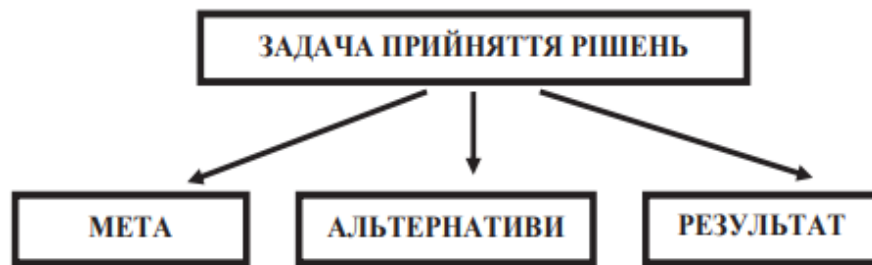


Рисунок 1.1 – Три складові задачі прийняття рішень

Узагальнену задачу прийняття рішень формально можна записати у вигляді:

$$\text{ЗПР} = \langle F, D, X, G, P \rangle,$$

де

- F – формулювання задачі, яка включає змістовний опис проблеми та за необхідністю її модельне представлення, визначення мети або сукупність цілей, які мають бути досягнуті, а також вимоги до вигляду кінцевого результату;

- D – сукупність можливих варіантів (альтернатив), з котрих проводиться вибір. Це можуть бути реально наявні варіанти (об'єкти, кандидати, способи досягнення мети, дії тощо), або гіпотетична множина всіх теоретично можливих варіантів, яка може бути навіть нескінченною;
- X – сукупність ознак (атрибутів), які описують варіанти та їх відмінності (особливості). В якості ознак виступають об'єктивні або суб'єктивні (експертні) показники, що характеризують альтернативи;
- P – переваги, які є основою для оцінювання та порівняння можливих варіантів рішення проблеми, відбору допустимих варіантів і пошуку найкращого або прийняттого варіанту;
- G – сукупність умов, що обмежують область допустимих варіантів розв'язку задачі. Обмеження можуть бути описані як змістовним чином, так і задані у вигляді деяких формальних вимог до варіантів або їх ознак. Наприклад, обмеження на значення будь-якої ознаки або неможливість одночасного поєднання певних значень ознак для реальних варіантів [4].

Задачу прийняття рішень можна формулювати у спрощеному вигляді

$$\text{ЗПР} = \langle D, O \rangle,$$

де D – множина варіантів (альтернатив), O – принцип оптимальності, який дає уявлення про якість варіантів або правило переваги варіантів.

Узагальнена схема прийняття рішень зводиться до виконання таких кроків:

- Крок 1. Аналіз поточної ситуації, тобто проблеми, яка потребує прийняття оптимального рішення.
- Крок 2. Прогноз розвитку цієї ситуації.
- Крок 3. Постановка задачі прийняття рішень (змістовна або формальна).
- Крок 4. Формування множини можливих варіантів d (альтернатив).

- Крок 5. Формування принципу О оптимальності рішень, зокрема, критеріїв для оцінки рішень.
- Крок 6. Розробка індикаторів для моніторингу реалізації окремих рішень d .
- Крок 7. Моделювання та порівняльна оцінка рішень d .
- Крок 8. Вибір найкращого (оптимального) рішення $* d$.
- Крок 9. Реалізація прийнятого рішення та його моніторинг.
- Крок 10. Загальне оцінювання результату вирішення проблеми.

У процесі прийняття рішень беруть участь особа (особи), яка приймає рішення, експерти та консультанти.

Особа, що приймає рішення (ОПР) – людина (або група людей), для якої вибір найкращої альтернативи слугує мотивом постановки задачі. ОПР має необхідні повноваження і несе відповідальність за прийняте рішення. Експерт – фахівець, який має інформацію про задачу, але безпосередньо не несе відповідальність за результат її вирішення. Експерт дає оцінки, необхідні для формування вихідної множини альтернатив і рішення задачі вибору.

Консультант – фахівець з теорії вибору та прийняття рішень. Він розробляє модель задачі, процедуру прийняття рішень, організовує роботу ОПР і експертів під час пошуку рішення. Консультантів також називають дослідниками, аналітиками або членами робочої групи [5].

Множина $D = \{d\}$ можливих альтернатив d може включати як незалежні, так і залежні альтернативи. Незалежними є ті альтернативи, будь-які дії з якими, наприклад, видалення з множини D , не впливають на якість інших альтернатив. У випадку залежних альтернатив оцінки одних з них впливають на якість інших.

Теорія прийняття рішень вивчає закономірності способів досягнення бажаного результату (мети) в умовах невизначеності різного типу, коли необхідно діяти в ситуації, що відома не повністю.

Можна виділити три групи невизначеностей:

невизначеність середовища, в якому приймають рішення;

невизначеність особи, що приймає рішення (ОПР), яка в загальному випадку може поводити себе непослідовно, бути суперечливою, допускати помилки, залежати від інших осіб (партнерів, суперників тощо), дії яких неможливо передбачити та повністю враховувати;

невизначеність цілей, які можуть не співпадати одна з одною.

У реальних ситуаціях часто важко або неможливо дати характеристику окремої альтернативи у вигляді числового критерія або сукупності критеріїв. Але, якщо розглядати альтернативу не окремо, а в парі з іншою, то знаходяться підстави сказати, яка з них краща (переважає за іншу). Не даремно народна мудрість говорить: «Все пізнається у порівнянні».

1.3 Огляд відомих методів ранжування

Для вирішення поставленої задачі, проведемо варіантний аналіз кількох методів, які можуть бути застосовані для ранжування.

Серед можливих варіантів розглянемо:

- Метод простого ранжування
- Метод безпосередньої оцінки
- Метод попарних порівнянь

1. Метод простого ранжування

Метод полягає в тому, що експерти розташовують об'єкти ранжування (наприклад, критерії) в порядку зниження, або зростання їх значущості (для альтернатив це спадання переваги) [6].

Ранжування дозволяє вибрати з досліджуваної сукупності факторів найсуттєвіший.

2. Метод безпосередньої оцінки

Метод полягає у віднесенні оціненого об'єкта до певного значення за оціночною шкалою (в присвоєнні об'єкту оцінки балів в певному інтервалі). Наприклад, оцінка від 0 до 10 згідно із перевагою по якійсь ознаці або їх групі (наприклад, альтернативі — за перевагою, критерію — за значущістю, фактором зовнішнього середовища — по впливу, проблемі — за пріоритетністю рішення).

Для подальшої обробки отримані оцінки можуть бути пронормовані, тобто їх сума може бути приведена до одиниці шляхом ділення кожної оцінки на їх загальну суму.

Для наочності і зручності обробки (здійснення вибору ОПР) оцінки можуть бути переведені в ранги. Максимальній оцінці при цьому відповідає найвищий ранг, тобто 1, а мінімальний — n (при n — кількості оцінених об'єктів).

3. Метод попарних порівнянь

Метод полягає у встановленні переваги об'єктів при порівнянні всіх можливих пар. Тут не потрібно, як при ранжируванні, впорядковувати всі об'єкти, необхідно в кожній з пар виявити більш значимий об'єкт або встановити їх рівність.

Парне порівняння являє собою процедуру встановлення уподобання об'єктів при порівнянні всіх можливих пар та подальше упорядкування об'єктів на основі результатів порівняння.

1.4 Опис моделі надання програмного забезпечення SaaS

SaaS - це модель надання ліцензії на програмне забезпечення за передплатою. SaaS розшифровується як software as a service - програмне забезпечення як послуга, також зустрічається переклад «інформація як сервіс».

SaaS найчастіше - це хмарні рішення, тобто ті що знаходяться на серверах в інтернеті.

Користувач отримує доступ до сервісу через браузер або по API. При цьому його підтримкою цілком займається постачальник послуги. Спрощено кажучи, модель SaaS це коли клієнт працює з готовим рішенням онлайн. Оплачує доступ і максимально швидко отримує на руки готовий інструмент.

Найпростіший приклад SaaS - це Google Docs, безкоштовний сервіс для роботи з документами. Ніяких носіїв, драйверів і установок. Переходите по посиланню і працюєте з текстами, таблицями та презентаціями прямо в браузері. Причому в документах одночасно з вами можуть працювати і інші колеги. Виїхавши у відрядження, можна зайти в свій аккаунт з будь-якого пристрою і продовжити друкувати потрібний документ. При цьому зберігати потрібно тільки налаштування, інші дані зберігаються автоматично.

По суті, SaaS-сервіс це єдине програмне ядро, яке надається в користування клієнтам. Доступ до системи вони отримують через мережу і можуть змінювати налаштування на свій розсуд. Обслуговуванням сервісу цілком займається провайдер послуги, а користувач тільки працює в ній.

Популярність цієї моделі щорічно зростає. Тільки в минулому році ринок SaaS збільшився на 21,7%. За прогнозами експертів, така тенденція збережеться і в найближчі роки. Це й не дивно, адже так користувачі отримують в розпорядження сучасні технології практично без зусиль зі свого боку.

Сьогодні з подібними рішеннями більшість людей стикається щодня. Поширене явище, коли SaaS-сервіси використовуються для бухгалтерського обліку, комунікації з клієнтами, редагування зображень, CRM, аналітики, ERP. Словом, практично для всього.

Переваги SaaS:

- У стандартній моделі відносини покупця і постачальника найчастіше закінчуються відразу після покупки ПО. А далі, як то кажуть, крутитесь, як

вмісте. Винятки становлять хіба що «гіганти» рівня Windows і Adobe. Коли ж мова йде про SaaS, користувач в будь-який момент може звернутися до постачальника за підтримкою. Всім клієнтам сервісу доступні регулярні оновлення.

- Для великої компанії це економія на ПО. Програму не потрібно купувати і окремо встановлювати на кожен комп'ютер, щоб в ній могло працювати кілька користувачів. Головне мати доступ до інтернету, а також логін і пароль.
- Деякі сервіси просто неможливо реалізувати на власному сервері компанії. Наприклад, продукти аналогічні Serpstat або Datanyze. Вони не можуть існувати в якості традиційного ПО, а тільки як SaaS. Таке рішення при великому бажанні можна розробити самостійно. Але для цього потрібні значні ресурси.

1.5 Аналіз існуючих програмних засобів попарного порівняння альтернатив

Першим аналогом є програма АНР Online System – BPMSG (див. рис. 1.2). Це програма мульти-критеріального прийняття рішень (Multi-criteria Decision Making Using the Analytic Hierarchy Process).

[AHP-OS Home](#) [Latest News](#) ([forgot?](#)) [Register](#)

AHP Priority Calculator

Try my free AHP online software [AHP-OS](#)

Select number of criteria and start pairwise comparisons to calculate priorities using the Analytic Hierarchy Process.

AHP Criteria

Select number of criteria:

Input number and names (2 - 20) OK

Pairwise Comparison Which university is better?

6 pairwise comparisons. Please do the pairwise comparison of all criteria. When completed, click *Check Consistency* to get the priorities.

Which criterion with respect to *Which university is better?* is more important, and how much more on a scale 1 to 9?

	A - Importance - or B?	or	B?	Equal	How much more?							
1	<input checked="" type="radio"/> VNTU		<input type="radio"/> Harvard	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input checked="" type="radio"/> 8	<input type="radio"/> 9
2	<input checked="" type="radio"/> VNTU		<input type="radio"/> KPI	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
3	<input checked="" type="radio"/> VNTU		<input type="radio"/> LP	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
4	<input type="radio"/> Harvard		<input checked="" type="radio"/> KPI	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
5	<input checked="" type="radio"/> Harvard		<input type="radio"/> LP	<input type="radio"/> 1	<input checked="" type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
6	<input checked="" type="radio"/> KPI		<input type="radio"/> LP	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9

CR = 0% Please start pairwise comparison

AHP Scale: 1- Equal Importance, 3- Moderate importance, 5- Strong importance, 7- Very strong importance, 9- Extreme importance (2,4,6,8 values in-between).

AHP-OS author: Klaus D. Goepel, BPMSG, [contact](#), last update: May 13, 2019

Рисунок 1.2 – Скріншот вікна програми AHP Online System – BPMSG

До плюсів програми можна віднести точні результати, можливість вирішення протиріч.

Мінусами є те що додаток має не зручний для користувача інтерфейс, та може бути складним для розуміння людині яка не є спеціалістом у теорії прийняття рішень.

Другим аналогом є програма Pairwise Comparison (див. рис. 1.3). Це android додаток який позиціонується як повсякденний додаток для вирішення питань вибору альтернатив.

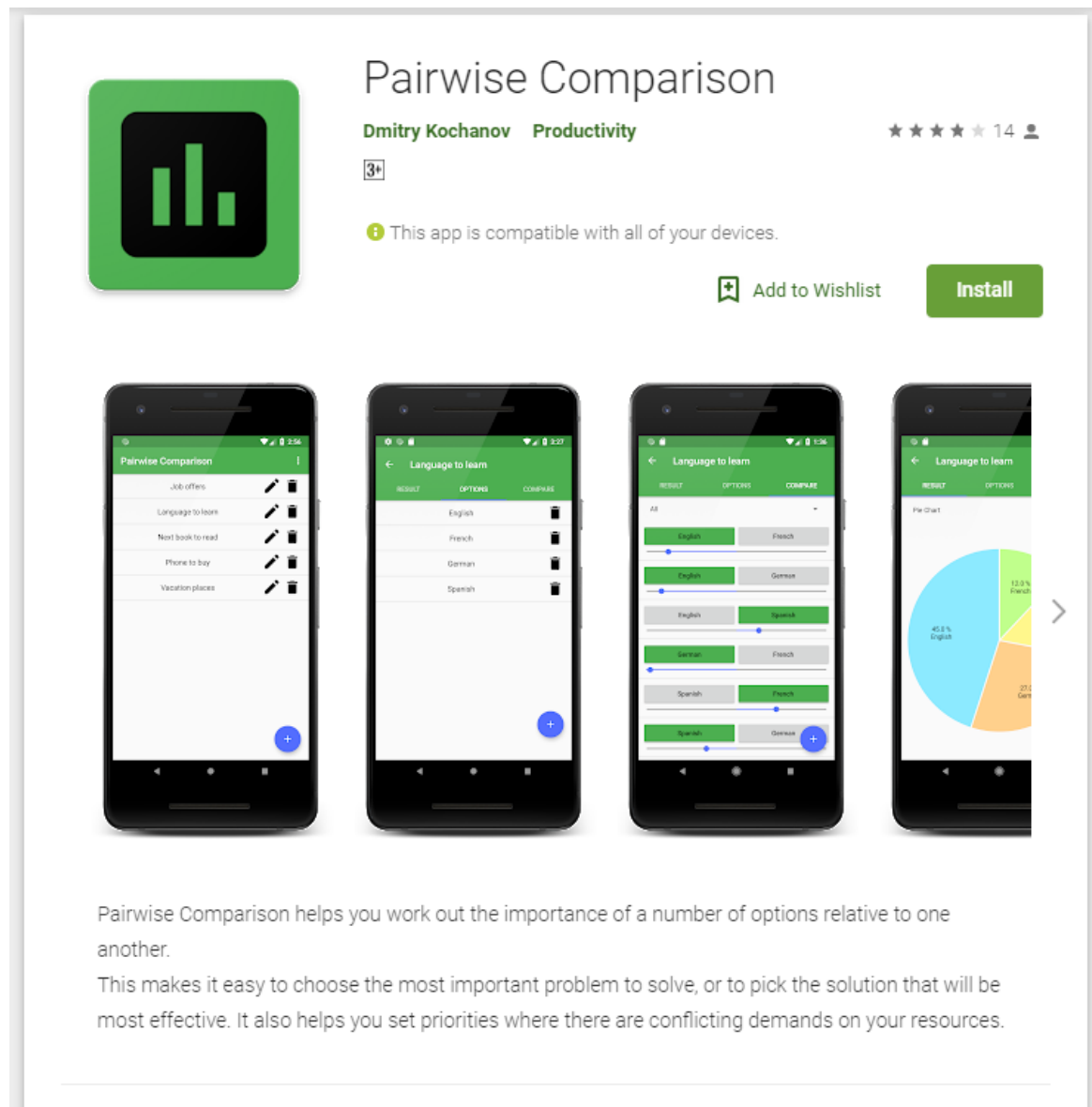


Рисунок 1.3 – Скріншот вікна програми АНР Online System – ВРMSG

Мінусами даної програми є те що вона судячи з відгуків часто працює з помилками, та має не зрозумілий інтерфейс.

1.6 Постановка задачі попарного порівняння альтернатив

В ході дослідження ми зосередимось на загальній системі попарних порівнянь, яка буде виконувати порівняння будь яких вхідних альтернатив.

Така система є досить універсальною, і може бути доналаштована для кожного конкретного клієнта, більш того має можливість бути використана як спеціалістами у теорії прийняття рішень, так і звичайними користувачами, наприклад, покупцями у магазині.

Модуль, що буде розроблений в ході дослідження, має бути представлений у вигляді веб-додатку і мати графічний інтерфейс.

Так як користувачу може не сподобатись наведений програмою набір альтернатив має бути можливість його редагування, для забезпечення точного порівняння

Для того щоб не спеціалісти могли зрозуміти принципи роботи програми має бути інструкція до програми, зображена у зручному вигляді.

Під час порівняння пари мають бути сформовані у випадковому порядку, для уникання ситуації коли коли користувач бачить одну і ту ж альтернативу декілька порівнянь підряд, особливо це буде актуально при порівняння бти і більше альтернатив.

Після введення користувач побаче результуюче ранжування. Результат буде представлено у вигляді повідомлення про найкращу альтернативу та графік графік з повними результатами.

Отже, в кінцевому підсумку, розроблений модуль забезпечить можливість на практиці показати результат проведеного в ході магістерської дипломної роботи дослідження.

1.7 Висновки

Розглянувши зазначені у попередньому розділі методи можемо стверджувати наступне стосовно нашої задачі:

Для програми яка націлена на широку аудиторію доцільніше обирати саме метод попарних порівнянь. Так як на відміну від ранжування, яке є більш

складним і його, швидше, застосовують при опитуваннях експертів, а не масових респондентів попарне порівняння зрозуміле для звичайного користувача.

Всі описані методи є досить ефективними та вартими уваги. Однак, враховуючи можливість максимально спростити та зробити зрозумілішою програму для користувача при використанні парного порівняння доцільно обрати цей метод.

2 ПРОЕКТУВАННЯ МОДУЛЮ ВІЗУАЛЬНОГО ПОПАРНОГО ПОРІВНЯННЯ З ПЕРЕДБАЧЕННЯ ВІДПОВІДІ

2.1 Структурна організація модулю візуального попарного порівняння з передбаченням відповіді

2.1.1 Функціональне моделювання системи

Функціональне моделювання за методологією SADT.

На початку моделювання потрібно визначити мету моделювання (Perpose), точку зору (ViewPoint), визначення моделі (Definition) та опис області її дії (Scope) [7].

Purpose: метою створення є процес моделювання модулю візуального попарного порівняння.

Viewpoint: при побудові моделі система була розглянута з точки зору розробника.

Definition: модель створюється для ілюстрації роботи системи на різних рівнях декомпозиції.

Scope: загальне функціонування модулю візуального попарного порівняння: формування альтернатив, ранжування альтернатив, та його результат.

Далі побудуємо модель, що буде відображає весь вказаний у завданні функціонал.

На рисунку 2.1 наведено контекстну діаграму процесу роботи модулю візуального попарного порівняння. Головний процес має зв'язок із зовнішніми

сутностями, де вхідними є вхідні дані користувача, зв'язок з сервером. Вихідними є результуюче ранжування.

Діаграма першого рівня декомпозиції, де робота модулю візуального попарного порівняння представлена у вигляді трьох наступних процесів: формування альтернатив, ранжування альтернатив, результуюче ранжування представлено на рисунку 2.2.



Рисунок 2.1 – Контекстна діаграма

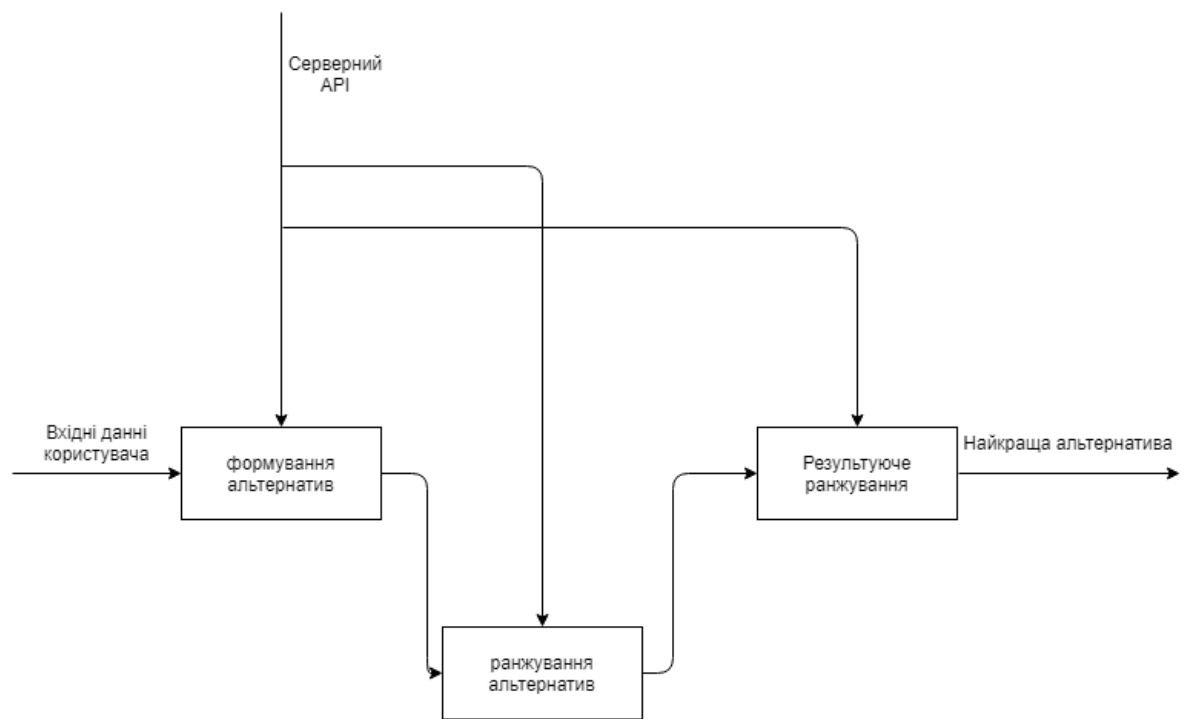


Рисунок 2.2 – Діаграма першого рівня декомпозиції

Діаграма другого рівня декомпозиції для блоку «Формування альтернатив», що включає в себе процес створення списку альтернатив, формування пар для порівняння, збереження пар складених з альтернатив які надіслав користувач на сервері зображено на рисунку 2.3.

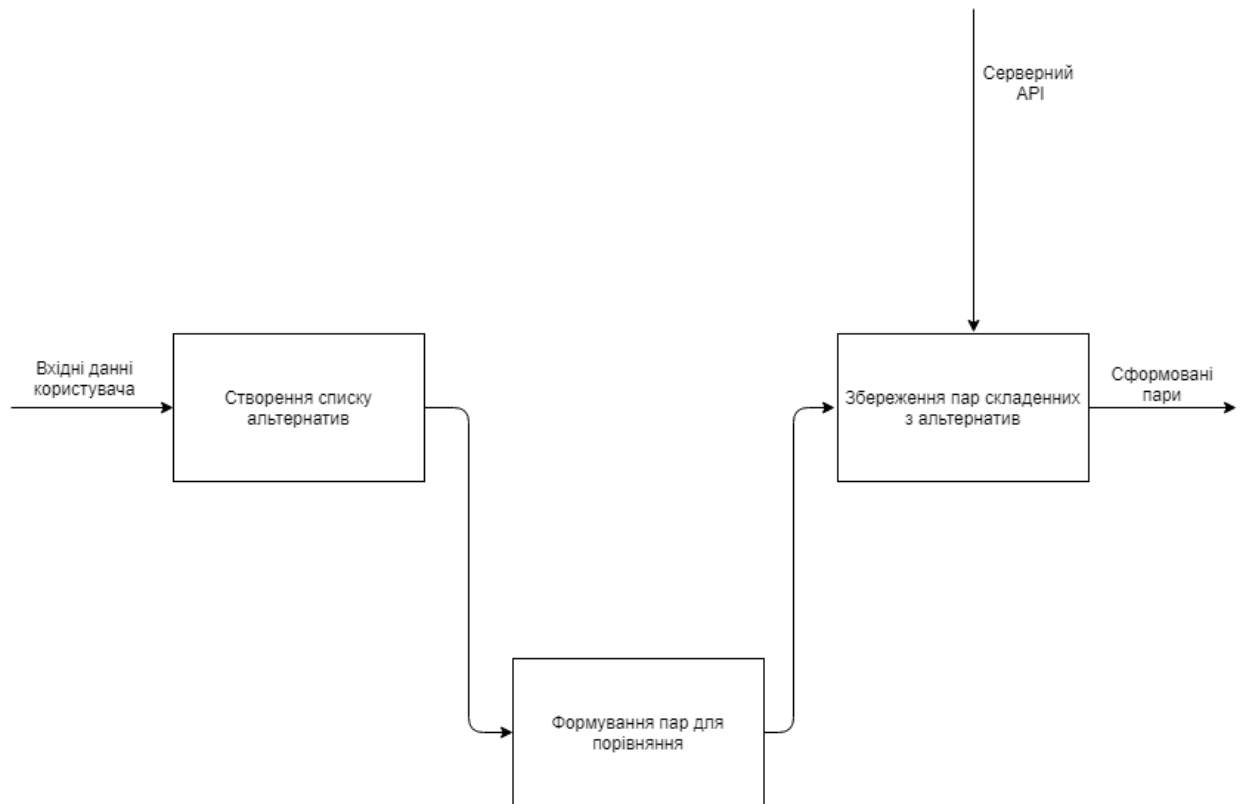


Рисунок 2.3 – Діаграма другого рівня декомпозиції для блоку «Формування альтернатив»

Діаграма другого рівня декомпозиції для блоку «Ранжування альтернатив», що містить в собі процес порівняння альтернатив, збереження результатів, формування результуючого ранжування, зображена на рисунку 2.4.

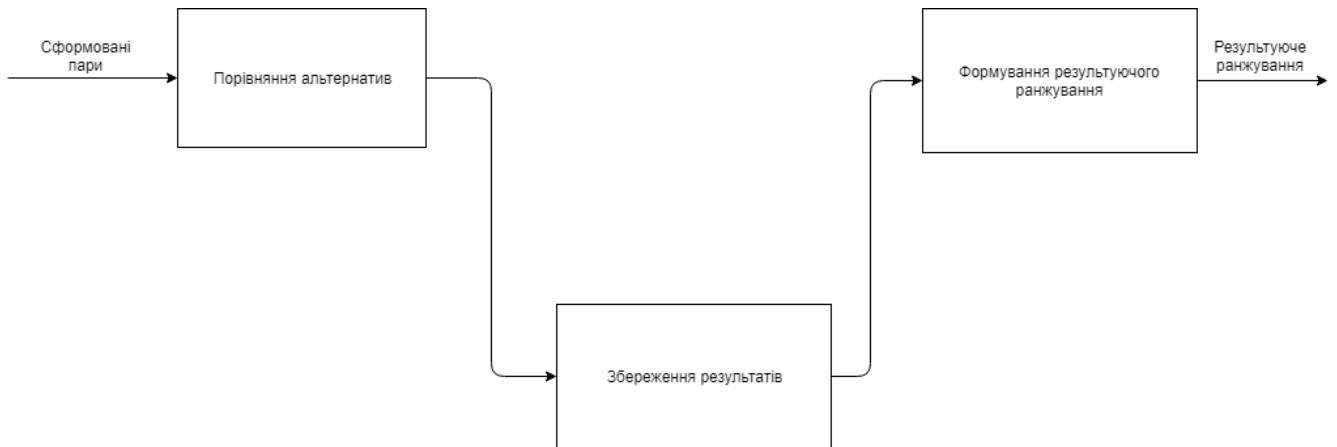


Рисунок 2.4 – Діаграма другого рівня декомпозиції для блоку «Ранжування альтернатив»

Діаграма другого рівня декомпозиції для блоку «Результуюче ранжування», що містить в собі процес формування запиту до серверу на відправку результату ранжування та формування графіку результуючого ранжування, зображена на рисунку 2.5.



Рисунок 2.5 – Діаграма другого рівня декомпозиції для блоку «Результуюче ранжування»

2.1.2 Проектування архітектури системи

Архітектура програмного забезпечення – це структура програми або обчислювальної системи, яка містить програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними [8].

Діаграми пакетів уніфікованої мови моделювання (UML) відображають залежності між пакетами, що становлять модель. Така діаграма дозволяє визначити архітектуру системи, встановивши залежності між програмними пакетами [9].

Нижче на рисунку 2.7 зображено діаграму пакетів, яка відображає структуру у вигляді пакетів головного компонента системи – модуля для клієнта. Діаграма містить такі пакети як App, View, Server, Modules.

App – основний пакет додатку, що формує та відображає інтерфейс додатку обробляє взаємодію користувача від етапу отримання альтернатив для ранжування до відправки на сервер.

View – даний пакет містить вхідну точку в програму та збирає всі елементи, які присутні в системі, наприклад деталі інтерфейсу, чи алгоритм результуючого ранжування.

Server – даний пакет реалізує розміщення веб-додатку у мережі інтернет, має утилітарну мету.

Modules – даний пакет містить утиліти для спрощення реалізації функціоналу системи.

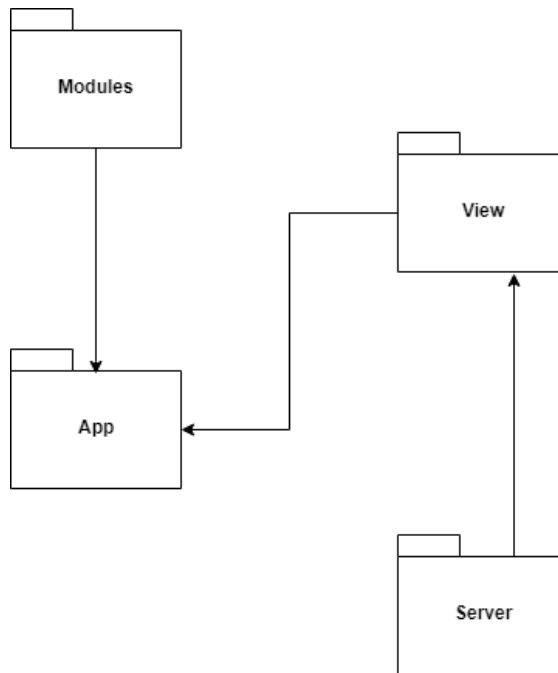


Рисунок 2.7 – Діаграма пакетів модуля для клієнта

Було проаналізовано діаграми пакетів та діаграми розгортання, відношення між їхніми елементами. Високий рівень абстракції не дозволяє зобразити деталізовані зв'язки, але це не заважає розумінню будови системи в цілому, загальна структура якої зображена на рисунку 2.8.

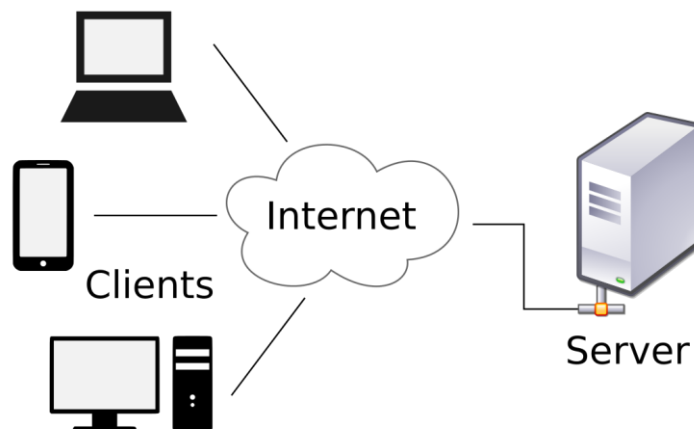


Рисунок 2.8 – Загальна структура системи

Було розроблено схему алгоритму функціонування модулю попарного порівняння (Рисунок 2.9).

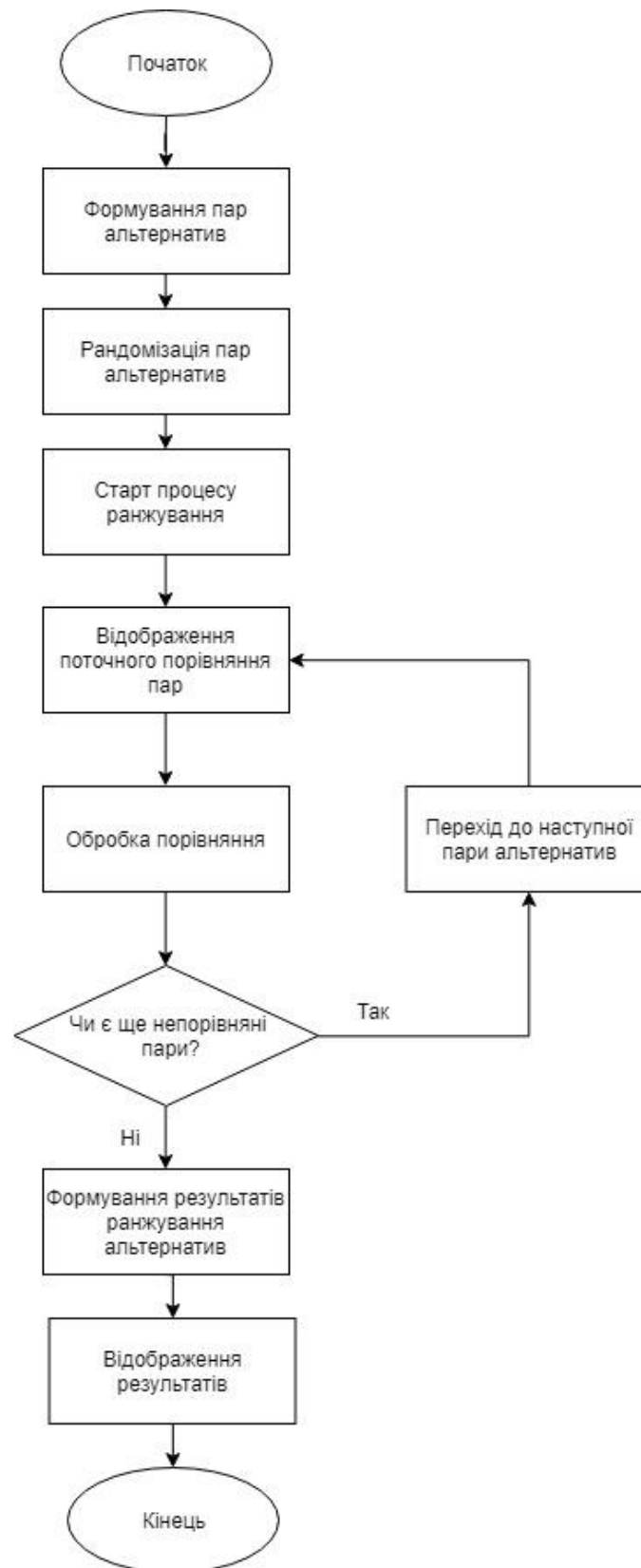


Рисунок 2.9 – Схема алгоритму функціонування модулю попарного порівняння

2.1.3 Проектування системи передбачення відповіді

Система передбачення відповіді складається з двох частин:

1. Очікувана відповідь – це відповідь яка обчислюється після порівняння більшості пар альтернатив, що є досить простим з точки зору логіки роботи, але зменшує кількість відмовлень користувача закінчити порівняння при великій кількості альтернатив.
2. Рекомендована відповідь – ця відповідь отримується за допомогою унікального алгоритму, що використовує статистичні дані відповідей користувачів. Алгоритм рекомендованої відповіді зображено на рисунку 2.10



Рисунок 2.9 – Схема алгоритму рекомендованої відповіді

Алгоритм працює наступним чином:

1. Отримання поточних результатів порівняння. На цьому етапі відбувається запит на сервер для отримання рекомендації в якому міститься дані про поточні результати
2. Відбір актуальних результатів – перебір статистичних даних та відсіювання альтернатив які вже містяться у порівнянні клієнта.
3. Створення мапи відповідей. Відбувається створення мапи альтернатив у вигляді { “Ім’я альтернативи”: {“Сума рейтингу”, “Кількість елементів суми”} }
4. Знаходження найкращої відповіді. Відбувається обчислення середнього рейтингу альтернатив та визначення найкращої рекомендації
5. Відправка рекомендації на клієнт.

2.2 Обґрунтування доцільності застосування попарних порівнянь для ранжування альтернатив

Метод парного порівняння - один з інструментів оцінки та вибору рішень, широко використовується в експертних оцінках при необхідності розставляти пріоритети в процесі будь-якої діяльності або ранжування різних об'єктів.

Ідея методу полягає в тому, що попарно порівнюються кожні два об'єкти і визначається першість одного з них, звідси назва - «попарне (або парне) порівняння». Вважається, що при вирішенні проблеми набагато легше зробити якісне порівняння двох об'єктів, спираючись на думку експертів, ніж встановити кількісні критерії. Строго кажучи, метод ПП - це метод отримання вихідних даних, метод своєрідного опитування респондентів. На базі отриманих даних

можна вирішувати різні завдання, зовсім необов'язково включати побудову оціночної шкали. Припустимо, що нас цікавить, як респонденти досліджуваної сукупності оцінюють будь-які об'єкти - професії, політичних лідерів, радіопередачі, якісь види товарів і т.д. Позначимо ці об'єкти через a_1, a_2, \dots, a_n (n - кількість оцінюваних об'єктів).

Фактор (об'єкт)	A_1	A_2	A_3	A_4	Сумма баллов
A_1	—	1	1	1	3
A_2	0	—	1	0	1
A_3	0	0	—	0	0
A_4	0	1	1	—	2

Таблиця 1 – приклад використання метода ПП

Розглянутий метод дозволяє отримати відповідь на це питання в досить своєрідному вигляді. Кожному респонденту пропонуються всілякі пари, складені з розглянутих об'єктів. Він повинен щодо кожної пари сказати, який об'єкт з цієї пари йому подобається більше. Скажімо, в разі розгляду в якості наших об'єктів деяких професій - наприклад, токаря, пекаря, лікаря і т.д. - ми запитуємо у кожного респондента, яка професія йому більше подобається: токаря або пекаря (фіксуємо відповідь), токаря або лікаря (фіксуємо відповідь), пекаря чи лікаря (фіксуємо відповідь) і т.д. для всіх можливих пар розглянутих об'єктів.

Основні переваги методу полягають в наступному:

- допускається вимір нерівномірної зміни важливості показників, так необхідне для вирішення більшості практичних економічних завдань (наприклад, покупка телефону);
- експерт в процесі аналізу зосереджує увагу не на всіх показниках відразу, а тільки на двох, порівнюваних в кожен даний момент, що полегшує роботу, а отже, сприяє підвищенню її якості;
- можна отримати велике число порівнянь кожного показника з іншими, завдяки чому підвищується точність оцінки і відкривається можливість

вивчати якість більшого числа сторін об'єкта дослідження, ніж при використанні інших методів;

- можна отримати не тільки середню оцінку показника, дану кожним експертом, але і дисперсію цієї оцінки, що дає можливість провести в подальшому більш глибокий економіко-математичний аналіз
- Метод парного порівняння - найбільш простий з існуючих класифікаційних тестів, оскільки він передбачає зіставлення тільки двох зразків продукту

Перевага методу парних порівнянь перед простим ранжуванням полягає в тому, що легше винести судження, оскільки супервізору одночасно потрібно порівнювати тільки двох чоловік. Друга перевага в тому, що він надає можливість поставити на один рівень людей, що володіють однаковими здібностями.

Метод парних порівнянь дозволяє провести строгий, статистично обґрунтований аналіз узгодженості думок експертів, виявити, випадкові чи ні отримані оцінки. Безсумнівно, процедура методу парних порівнянь складніше методу простого ранжування, але простіше методу послідовних порівнянь.

Даний метод дуже простий і він дозволяє досліджувати більшу кількість об'єктів (у порівнянні, наприклад, з методом рангів) і з більшою точністю.

Основні недоліки методу полягають в наступному:

в необхідності виконувати величезну кількість парних порівнянь, якщо доводиться оцінювати великі групи. Супервізор, у якого в підпорядкуванні знаходиться 60 співробітників, повинен буде виконати 1770 порівнянь! Якщо ж порівняння буде проводитися по п'яти окремими параметрами, то кількість парних порівнянь зросте в п'ять разів. Зазвичай методом парних порівнянь користуються в двох випадках: або при проведенні оцінки невеликих груп, або

при проведенні оцінки тільки по одному параметру - по загальній ефективності виробничої діяльності

Один недолік парних порівнянь як експериментального методу полягає в тому, що порівняння п стимулів вимагає отримання $(p - 1) \times (n / 2)$ суджень. Напр., Для 10 стимулів потрібно отримати 45 суджень, а якби ми захотіли шкалу з 50 стимулів, нам треба було б +1225 суджень.

Основний мінус методу парних порівнянь - немає готових програмних розробок.

Щоб скористатися цим методом, доведеться доручити компетентним працівникам компанії розробити спеціальну програму або замовляти її у стороннього розробника. Крім того, поки не існує апробованих бізнес-рішень в галузі ділової оцінки, заснованих виключно на парних порівняннях, це скоріше математичний напівфабрикат, який служить для створення інструментів шкалювання. До того ж сама тестова процедура виглядає досить монотонної, що впливає на об'єктивність дослідницьких висновків. Та й якість результатів безпосередньо залежить від кількості оцінок і оцінюваних показників, а також правильного відбору пар і їх однозначного тлумачення.

Недоліком методу є зростання трудомісткості процедури при збільшенні числа об'єктів: вже при 12-15 об'єктах процедура стає трудомісткою. Крім того, різні пари об'єктів іноді зіставляються респондентами за різними критеріями, що призводить до нетранзитивності переваг. Метод широко використовується в експертних оцінках.

Але майже всі мінуси зводяться до мінімуму у проектуємій системі так як зіставляння по різним критеріям не можливо, так як респонденти попередню обирають для себе в системі найважливіші критерії, а кількість можливих об'єктів ранжування буде мінімальною через відкидування завідомо найгірших по критеріальній оцінці.

2.3 Висновки

В даному розділі проведено розробку структури модулю візуального попарного порівняння, функціональне моделювання за методологією SADT, промодельовані інформаційні потоки сценаріїв, та побудовано діаграму першого рівня декомпозиції, де робота модулю візуального попарного порівняння представлена у вигляді трьох наступних процесів: формування альтернатив, ранжування альтернатив, результуюче ранжування, за допомогою чого було відображено структурну організацію модулю візуального попарного порівняння, та зображено загальну схему системи.

Також було обґрунтовано доцільність застосування методу попарного порівняння для ранжування альтернатив. Адже у даній магістерській дипломній роботі буде реалізовано функціонал, який дозволить надавати користувачам можливість обрати найкращу для них альтернативу максимально швидко.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЮ ВІЗУАЛЬНОГО ПОПАРНОГО ПОРІВНЯННЯ

3.1 Обґрунтування вибору технологій для розробки

Для реалізації додатку необхідно було обрати Front-end фреймворк та бібліотеки для реалізації користувацького інтерфейсу, та взаємодії з серверною частиною.

Фундаментальним елементом сучасної frontend розробки є javascript фреймворк. Для реалізації даного додатку було розглянуто наступні фреймворки: Angular, Vue, React.

React не відноситься до фреймворків в чистому вигляді. Це свого роду модифікована бібліотека, «заточена» під MVC (Model-View-Controller, де Модель відповідає за надання даних, Вид - відображає дані Моделі користувачеві, а Контролер інтерпретує дії користувача і змушує Модель вносити зміни) [10].

Angular [11] і Vue.js [12] відносяться вже до самих фреймворків.

Якщо в основі архітектури проекту лежить React, то це дозволяє:

- Зменшити розмір скомпільованого додатку, що дуже критично при імплементації додатку у вигляді підключаемого компонента;
- Більше можливостей для нестандартних програмних рішень;
- Легкість додавання компоненту у існуючі проекти незалежно від їх програмної реалізації.

Отже, за допомогою React можливо задати більш гнучку архітектуру та реалізувати будь який API (Application Programming Interface - набір готових команд).

У разі використання готових фреймворків - Vue.js і Angular, проблем з підбором або налаштуванням бібліотек для різних задач не виникає але ці фреймворки містять багато зайвого, та мають великий розмір для рішення поставленої задачі.

Це дозволить підключитися до проекту сторонньому програмісту без тривалого вивчення архітектури додатку. Саме в уніфікації процесів криється популярність повноцінних фреймворків.

React і Vue.js підтримують тільки односторонню передачу даних. При цьому в React об'єкти вміщені. Говорячи простою мовою, кожен з об'єктів програми відноситься до кінцевих процедур, які не вимагають дій користувача до закінчення роботи.

Однак React підтримує копіювання і передачу стану. Тобто, властивості прописаних об'єктів можуть бути відновлені на іншому пристрої, якщо запустити додаток і повідомити стан компонентів. Отже, рендеринг буде ідентичним, на екранах обох пристроїв буде одна і та ж «картинка».

Фреймворк Vue.js працює вже трохи інакше. JS і раніше односторонній, але компоненти працюють з шаблонами, і на виході виходить чистий html. Є підтримка JSX, що спрощує перехід з React і схожих бібліотек.

Для фреймворка автоматизована логіка процесів за рахунок побудови директив. Досить додати потрібну директиву, щоб отримати додаткову функціональність. Наприклад, необхідно зробити блоковий елемент

переміщуються по сторінці. Для реалізації досить додати директиву «draggable» до властивостей об'єкта, після чого його можна буде перетягувати мишкою по екрану.

Фреймворк Angular дещо відрізняється логікою процесу. Так, тут є все, що властиво Vue.js, однак опис взаємодії об'єктів відбувається в службах, які є складовими частинами модулів.

Модульна архітектура більш зручна при розробці великих додатків. Модуль створюється для вирішення декількох завдань схожою функціональністю. Отже, кінцевий розмір коду стає менше, а швидкість обробки вище.

Підтримка MVVM (Model-View-ViewModel) [13] допускає рішення різних завдань в одному розділі програми з використанням одного набору даних. Залежність функцій визначає двосторонню спрямованість передачі даних. Кожна процедура може запустити інший процес.

Для реалізації додатку було обрано React через гнучкість та легкість використання у вигляді standalone-компонента швидко реалізувати додаток не жертвуючи при цьому якістю реалізації.

3.2 Основні оператори мови програмування javascript

Основні оператори мови програмування, використані в процесі розробки програми:

- оператор вибору if;
- оператор вибору switch;
- оператор циклу for;
- оператор циклу for of
- оператор циклу while.
- оператор bind

Оператор вибору if дозволяє нам виконувати чи не виконувати певні інструкції, в залежності від того чи є істинним чи хибним умова цього оператора. Одне з найважливіших призначень оператора вибору if так це те, що він дозволяє програмі зробити дію на вибір, в залежності від того, які дані запровадив користувач.

За допомогою конструкції `switch` здійснюється вибір кількох альтернатив, в залежності від значення змінної. Якщо знаходяться збіги, виконується відповідні команди до стрічки `break`, або до кінця оператора. Якщо ж збігів не знайдено – виконується команда після ключового слова `default`.

Цикл з лічильником (`for`) — цикл, в якому деяка змінна змінює своє значення від заданого початкового значення до кінцевого значення з деяким кроком, і для кожного значення цієї змінної тіло циклу виконується один раз. В більшості процедурних мов програмування реалізується оператором `for`, в якому вказується лічильник (так звана «змінна циклу»), потрібна кількість проходів (або граничне значення лічильника) `i`, можливо, крок, з яким змінюється лічильник.

`For of` - забезпечує простий і зрозумілий спосіб ітерації елементами масиву або будь-якої лічильної колекції. Оператор `foreach` обробляє елементи в порядку, повернутому масивом або переліку типу колекції, зазвичай від нульового до останнього елемента. Наприклад, наступний код створює масив `numbers` і здійснює ітерацію по ньому за допомогою оператора `for of`.

Цикл `while` виконується до тих пір, поки керуючий логічний вираз є істиною

Метод `bind()` створює нову функцію, яка в момент виклику має певне присвоєне значення `this`, а також задану послідовність аргументів, що передують будь-яким аргументам, переданим під час виклику нової функції.

3.3 Обґрунтування вибору середовища розробки

При виборі середовища розробки потрібно опиратися на операційну систему встановлену на вашому комп'ютері та можливості середовища розробки.

Під час розробки програмного додатку було розглянуто такі середовища розробки:

Sublime Text 3 - швидкий кросплатформовий редактор початкових текстів програм. ST3 не є вільним чи відкритим програмним забезпеченням, але деякі його плагіни розповсюджуються з вільною ліцензією, розробляються і підтримуються спільнотою розробників. В даному середовищі можливе покрокове тестування системи та окремих модулів, що значно зменшує можливість виникнення критичної помилки на кінцевому етапі реалізації системи та дозволяє коригувати реалізацію, покращуючи систему після кожного набору сценаріїв. [14]

Можливості:

- автозбереження
- закладки в файлах
- мінімапа: попередній перегляд повного початкового тексту
- редагування на кількох панелях
- вбудована підтримка 27 мов програмування
- підсвітка синтаксису, що повністю налаштовується
- відповідність дужок, автодоповнення
- підтримка макросів та плагінів
- користувацькі гарячі клавіші
- функція швидкого переходу до певної ділянки коду
- надання швидкого доступу до команд

Notepad++ — текстовий редактор, призначений для програмістів і тих, кого не влаштовує скромна функціональність Блокнота, що входить до складу Windows. Notepad++ базується на компоненті Scintilla (потужному компоненті для редагування), написаному на C++ з використанням тільки Win32 API і STL, що забезпечує максимальну швидкість роботи при мінімальному розмірі

програми. Інтерфейс у Notepad++ — багатомовний. Серед особливостей програми — підсвічування синтаксису, підтримка великої кількості мов (C, C++, Java, XML, HTML, PHP, Java Script, ASCII, VB/VBS, SQL, CSS, Pascal, Perl і Python), багатомовна підтримка, робота з декількома документами. Підсвічування тексту і можливість згортання блоків, згідно з синтаксисом мови програмування WYSIWYG (друкуєш і отримуєш те, що бачиш на екрані) Режим підсвічування синтаксису, що налаштовується користувачем Авто-завершення слова, що набирається Одночасна робота з безліччю документів Підтримка регулярних виразів для пошуку/заміни Повна підтримка перетягування фрагментів тексту Динамічна зміна вікон перегляду Автоматичне визначення стану файлу Підтримка великої кількості мов Замітки Плагіни Запис макросу і його виконання

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. У Microsoft Visual Studio 2015 присутня низка нових функцій: розробки для Windows 8, Інтернету, SharePoint, мобільних пристроїв і хмарних служб, а також засобу управління життєвим циклом додатків, що дозволяють усунути перешкоди між співробітниками і скоротити цикли для безперервного нарощування цінності кінцевого результату. [15]

Порівняння середовищ розробки можна побачити на таблиці 3.1.

	ST3	VS 2015	Notepad ++
Інтелектуальна технологія	+	+	-
Підтримка авто запуску	+	+	+
Дебаг коду	+/-	+	+/-
Інструменти обробки	+	+	+
Автоконтроль	+	+	+
Інтеграція з іншими мовами	+	+/-	+

Таблиця 3.1 – Порівняння середовищ розробки

Отже через свої переваги, в даному проєкті буде використано Visual Studio Code, а саме через можливе покрокове тестування системи та окремих модулів, що значно зменшує можливість виникнення критичної помилки на кінцевому етапі реалізації.

3.4 Тестовий приклад роботи програми і аналіз результатів

Для перевірки програмного модуля на працездатність проведемо його тестування. Початкове вікно програми та вікно входу в систему зображено на рисунку 3.1.

Початкове вікно отримує сформовані альтернативи для порівняння.

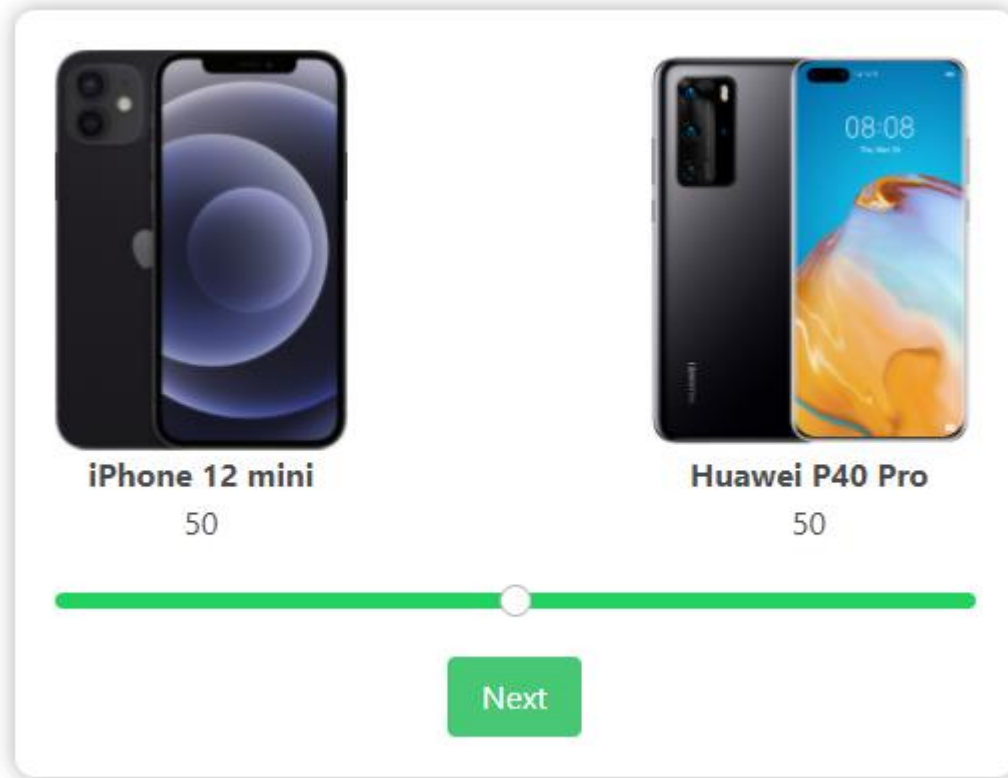


Рисунок 3.1 – Початкове вікно

Порівнювати альтернативи можливо за допомогою пересування повзунка, що зображено на рисунку 3.3.

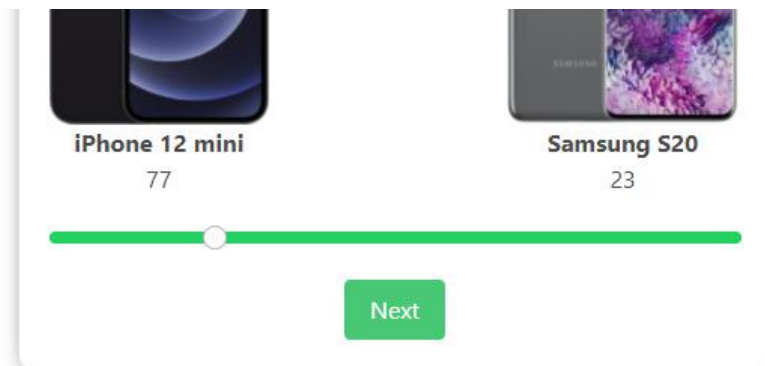


Рисунок 3.3 – Процес порівняння

Використовуючи повзунок шляхом його пересування вправо або вліво необхідно порівнювали альтернативи. При пересуванні повзунка відображаються одиниці які отримає та чи інша альтернатива.

Після отримання рекомендованої альтернативи, відповідний блок з'являється у нижній частині. Вікно додатку з рекомендованою альтернативою зображено на рисунку 3.4.

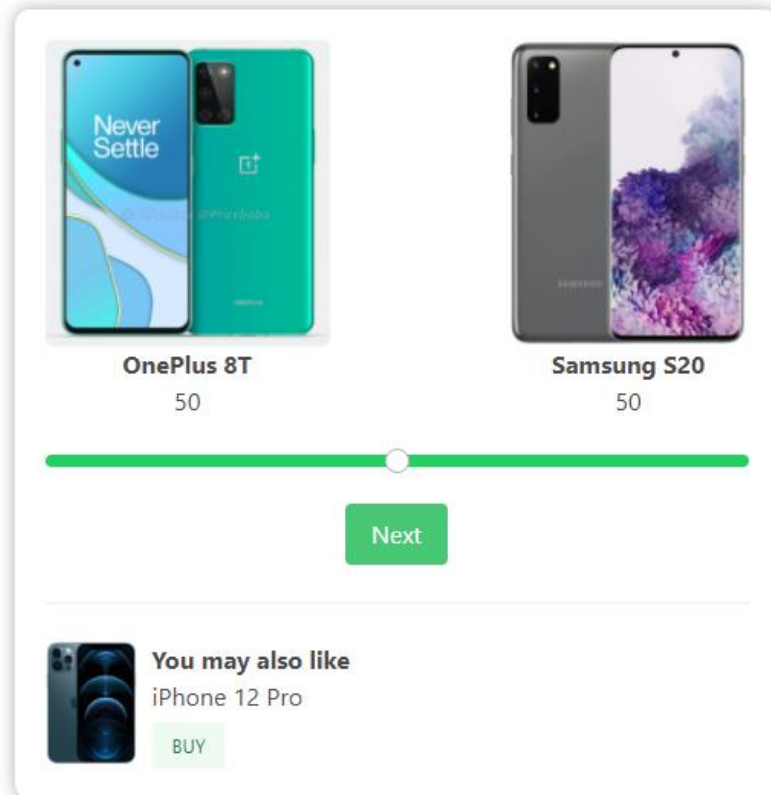


Рисунок 3.4 – Приклад порівняння альтернатив

Також бачимо логи сервера, що було знайдено 3 рекомендації та обрано найкращу. Логи сервера зображено на рисунку 3.5.

```
{
  'iPhone 12 Pro': { rate: 142, i: 4 },
  'OnePlus 8T': { rate: 17, i: 2 },
  'Samsung Galaxy Note 20 Ultra': { rate: 7, i: 1 }
}
Best match: { name: 'iPhone 12 Pro', rate: 142, i: 4 }
{
```

Рисунок 3.5 – Логи сервера

Після порівняння половини пар альтернатив, з'являється блок з передбаченою відповіддю, що зображено на рисунку 3.5.

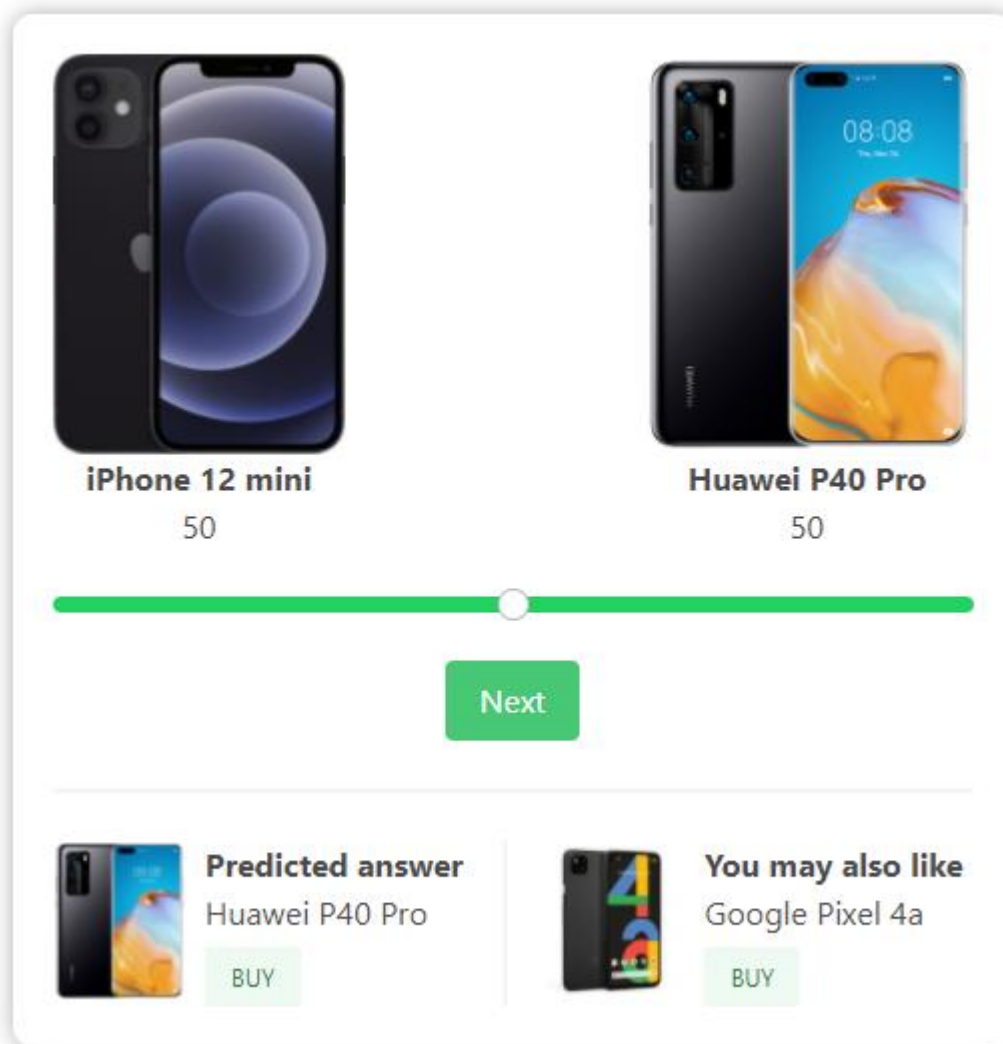


Рисунок 3.5 – Передбачена відповідь

Після завершення процесу ранжування відображаються результати та найкраща альтернатива. Результати ранжування зображено на рисунку 3.5.

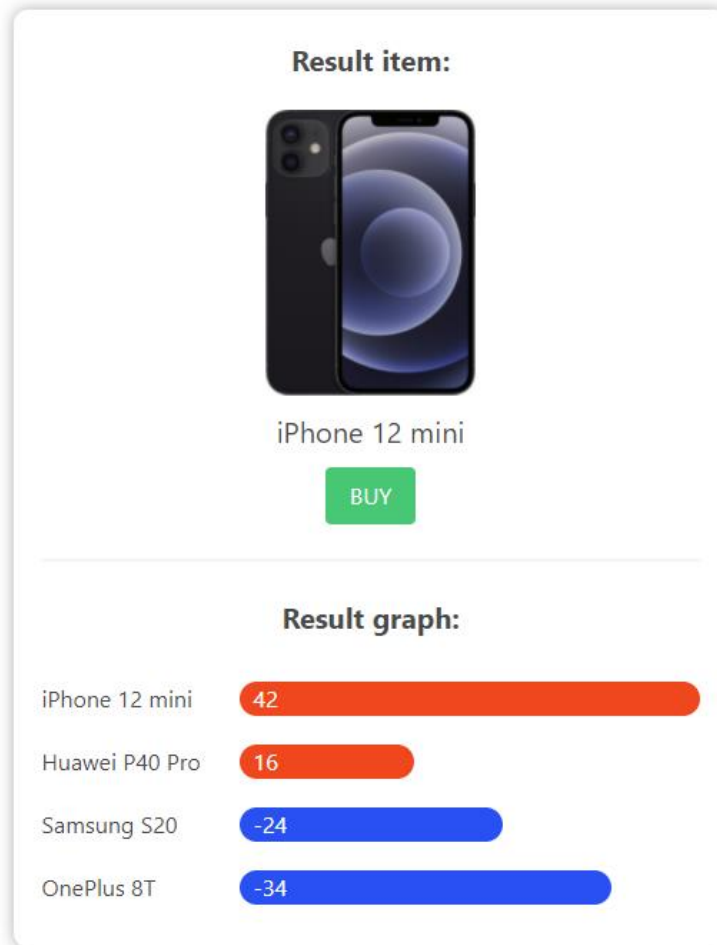


Рисунок 3.5 – Результати ранжування

3.5 Висновок

У даному розділі було обрано React бібліотекою для front-end розробки.

Виконано аналіз існуючих найпопулярніших середовищ програмування для обраної мови javascript та платформи було обрано середовище програмування Visual Studio Code, яке задовольняє усім потребам при розробці програмного забезпечення, має зручний графічний інтерфейс а також засоби налагодження.

Було проведено тестування програмного модуля на виконання його основної функції – попарне порівняння альтернатив.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть к.т.н., доц. кафедри КН Колодний В. В. та к. т. н. ст. викладач кафедри КН Озеранський В. С.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали експерта	
	1. Колодний В. В.	2. Озеранський В. С.
	Бали, виставлені експертами:	
1	4	4
2	4	3
3	5	4
4	4	4
5	4	4
6	3	4
7	4	3
8	4	5
9	4	5
10	4	3
11	4	4
12	4	4
Сума балів	СБ ₁ = 48	СБ ₂ = 47

Середньоарифметична сума балів \overline{CB}	$\overline{CB} = \frac{\sum_1^3 CB_1}{2} = 47.5$
---------------------------------------------------	--------------------------------------------------

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту потрібні наступні витрати.

Основна заробітна плата для розробників визначається за формулою:

$$Z_o = \frac{M}{T_p} * t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника;

T_p – кількість робочих днів у місяці, $T_p = 22$ дні;

t – число днів роботи розробника, $t = 40$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	9000	409,09	8	3272,72
Інженер-програміст	7500	340,90	40	13636
Всього				16908,72

Розрахуємо додаткову заробітну плату:

$$З_{\text{дод}} = 0,1 \cdot 16908,72 = 1690,87 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 22% від суми їхньої основної та додаткової заробітної плати:

$$Н_{\text{зп}} = (З_о + З_р) * \frac{\beta}{100}, \quad (4.2)$$

$$Н_{\text{зп}} = (16908,72 + 1690,87) * \frac{22}{100} = 4091,90 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц * Н_a}{100} * \frac{T}{12}, \quad (4.3)$$

де Ц – балансова вартість обладнання, грн;

Н_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання (T = 2 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	20000	25	2	833.33
Всього				833.33

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n N_i * Ц_i * K_i, \quad (4.4)$$

де n – кількість комплектуючих;

N_i – кількість комплектуючих i-го виду;

Ці – покупна ціна комплектуючих і-го виду, грн;

Кі – коефіцієнт транспортних витрат (прийmemo Кі = 1,1).

Таблиця 4.4 – Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	200	1	200
Пачка паперу	уп.	120	1	120
Ручка	шт.	5	1	5
Всього з урахуванням транспортних витрат				362

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V * P * \Phi * K_{\text{п}}, \quad (4.5)$$

де V – вартість 1кВт-години електроенергії (V=1,7 грн/кВт);

P – установлена потужність комп'ютера (P=0,8кВт);

Φ – фактична кількість годин роботи комп'ютера (Φ=200 год.);

K_п – коефіцієнт використання потужності (K_п< 1, K_п = 0,7).

$$V_e = 1,7 * 0,6 * 200 * 0,7 = 190.4 \text{ (грн)}$$

Розрахуємо інші витрати V_{ін}.

Інші витрати I_в можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які виконували дану роботу, тобто:

$$V_{\text{ін}} = (1..3) * (Z_o + Z_p). \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$V_{\text{ін}} = 1 * (8363,36 + 836,33) = 9199,69 \text{ (грн).}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$V = Z_o + Z_d + H_{\text{зн}} + A + K + V_e + I_b$$

$$V = 16908,72 + 1690,87 + 4091,90 + 833,33 + 190,4 + 362 + 9199,69 = 33277,98 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи за формулою:

$$V_{\text{заг}} = \frac{V_{\text{ін}}}{\alpha}, \quad (4.7)$$

де – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$V_{\text{заг}} = \frac{33277,98}{1} = 33277,98$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\beta}, \quad (4.8)$$

де – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{33277,98}{0,9} = 36974,87 \text{ (грн)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} * N + \Pi_{\text{я}}\Delta N)_i a, \quad (4.9)$$

де $\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 25 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 25 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на

500 користувачів, протягом другого року – на 600 користувачів, протягом третього року – 1000 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 1000 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 100 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 25 * 700 + (100 + 25) * 500 = 80000 \text{ грн.}$$

Протягом другого року:

$$\Delta\Pi_2 = 25 * 700 + (100 + 25) * (500 + 600) = 155000 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = 25 * 700 + (100 + 25) * (500 + 600 + 1000) = 280000 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - \text{PV}), \quad (4.10)$$

де ПП – вартість чистих прибутків;

PV – теперішня вартість інвестицій $PV = \text{ЗВ}$, грн.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.



Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_1^m \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

m – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0, 1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{36974,87}{(1+0,1)^0} + \frac{80000}{(1+0,1)^2} + \frac{155000}{(1+0,1)^3} + \frac{280000}{(1+0,1)^4} = 413189,88 \text{ (грн)}$$

Розрахуємо $E_{\text{абс}}$:

$$E_{\text{абс}} = 413189,88 - 36974,87 = 376215.01 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$ за формулою:

$$E_{\text{в}} = \sqrt[T]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (4.12)$$

де $E_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = ZB$, грн;

T – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_{\text{в}} = \sqrt[T]{1 + \frac{376215.01}{36974,87}} - 1 = 1.22 \text{ або } 122 \%$$

Далі, розраховану величину $E_{\text{в}}$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,1 + 0,1 = 0,2$$

Оскільки $E_B = 122\% > \tau_{\text{мін}} = 0,2 = 22\%$, то й інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}$$

Отже, розрахуємо термін окупності вкладених інвестицій:

$$T_{\text{ок}} = \frac{1}{1,22} = 0,81 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновок

В даному розділі було проведено економічне обґрунтування доцільності розробки програми для попарного порівняння з передбаченням відповіді. Незалежними експертами було здійснено оцінювання комерційного потенціалу розробки, за результатами якого було визначено, що нова розробка має високий рівень комерційного потенціалу, оскільки середньоарифметична сума балів становить 47.5. Також було виконано прогнозування витрат на виконання розробки, де розраховано основну заробітну плату кожного із розробників, додаткову заробітну плату всіх розробників, нарахування на заробітну плату, амортизацію обладнання, комп'ютерів та приміщень, витрати на допоміжні матеріали, витрати на силову електроенергію тощо. Загальна сума витрат на виконання означених робіт склала 36974,87 грн. Виконано розрахунок ефективності вкладених інвестицій та періоду їх окупності. Визначено, що абсолютна ефективність вкладених інвестицій становить 376215.01 грн, і це свідчить про те, що вкладання коштів на виконання та впровадження результатів

НДДКР є доцільним. Було розраховано відносну (щорічну) ефективність вкладених в наукову розробку інвестицій – 122 %, її величина більша за мінімальну (бар'єрну) ставку дисконтування, отже інвестор буде зацікавлений у фінансуванні даної наукової розробки. Проведено розрахунок терміну окупності - 0,81 року (10 місяців). Це означає, що вже починаючи з 11 місяця розробка буде приносити прибуток. Взагалі можна зробити висновок, що фінансування розробки програми для попарного порівняння з передбаченням відповіді є економічно доцільним проектом.

ВИСНОВКИ

Під час виконання магістерської дипломної роботи був проведений аналіз предметної області ранжування альтернатив. Був проведений аналітичний огляд методів-аналогів для розв'язання задачі ранжування, який показав доцільність використання саме попарного ранжування.

Реалізовано програмний засіб для попарного ранжування альтернатив, детально описано програмний модуль, що відповідає за роботу алгоритму ранжування. Розроблена програма реалізує зручний спосіб попарного ранжування альтернатив.

Тестування розробленого модуля для ранжування альтернатив показало його працездатність. Програмний засіб повністю відповідає вимогам, поставленим перед ним на початку дослідження – зручність процесу ранжування для користувача.

В результаті виконання даної магістерської дипломної роботи поставлені задачі були виконані в повній мірі.

Отже, мета магістерської роботи досягнута.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ВНТУ – [Електронний ресурс]. – Режим доступу: <http://conf.vntu.edu.ua/allvntu/2018/>
2. А. Д. Хохлов, В. В. Колодний, «Комп’ютерна програма “Інформаційна технологія візуального моделювання попарних порівнянь» (ВМПП)” Свідоцтво про реєстрацію авторського права на твір № 79226. Дата реєстрації 22.05.2018.
3. О.І. Кушлик-Дивульська, Б.Р. Кушлик. Основи теорії прийняття рішень. – Київ, НТУУ «КПІ», 2014. – 3с.
4. Теорія прийняття рішень – КПІ [Електронний ресурс]. – Режим доступу: URL: <http://mses.kpi.ua/knigi/zmist/TPR.pdf>. - Назва з екрану.
5. С. В. Зибін. Сучасна спеціальна техніка — М.: Державний університет телекомунікацій, 2017. — С. 44. — , ISSN 2411-3816.
6. Методи обробки експертної інформації – Вікіпедія [Електронний ресурс]. - Режим доступу: URL: https://uk.wikipedia.org/wiki/Методи_обробки_експертної_інформації - Назва з екрану.
7. Методи структурного аналізу і проектування [Електронний ресурс]. – Режим доступу: <http://or-rsv.narod.ru/SADT/SADT.htm>
8. Архітектура програмного забезпечення – [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org/wiki/Архітектура_програмного_забезпечення
9. Буч Г. Язык UML: руководство пользователя. Пер. с англ. [Текст] / Г. Буч, Д. Рамбо, А. Джекобсон. – М.: ДМК, 2000. – 457 с.
10. MVC Design Pattern – [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/mvc-design-pattern/>
11. Angular – [Електронний ресурс]. – Режим доступу: <https://angular.io/>

12. Vue.js – [Электронный ресурс]. – Режим доступа: <https://vuejs.org/>
13. MVVM: полное понимание (+WPF) Часть 1 – [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/338518/>
14. Documentation – Sublime Text – [Электронный ресурс]. – Режим доступа: <https://www.sublimetext.com/docs/3/>
15. Visual Studio Code – Code Editing. Redefined – [Электронный ресурс]. – Режим доступа: <https://code.visualstudio.com/>