

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

ДОПУСКАЄТЬСЯ ДОЗАХИСТУ

:

Завідувач кафедри

Комп'ютерних систем, мереж та
кібербезпеки _____

Касаткін Д. Ю., к. п. н., доц.

підпис ПІБ, вчене звання і ступінь

«__» _____ 2025 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

На тему: Розробка системи підтримки прийняття рішень при виборі
хмарних сервісів для організацій

Спеціальність 123 «Комп'ютерна інженерія»

Гарант освітньої програми

к.фіз.-мат.н., доц.

(підпис)

/ Нікітенко

(ПІБ)

Керівник дипломного проекту: _____ / Нікітенко Є.В. /

(підпис)

(ПІБ)

Виконав: _____ / Штепа Д.Д. /

(підпис)

(ПІБ)

КИЇВ–2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз вимог до системи	20. 03. 2025	Виконано
2	Проектування системи	01. 04. 2025	Виконано
3	Реалізація системи	12. 04. 2025	Виконано
4	Тестування системи	14. 04. 2025	Виконано
5	Оформлення пояснювальної записки	18. 05. 2025	Виконано
6	Оформлення графічного матеріалу	19. 05. 2025	Виконано

Студент

Штепа Д.Д.

_____ (підпис)

_____ (ініціали та прізвище)

Керівник проекту (роботи)

_____ (підпис)

Нікітенко Є.В.

_____ (ініціали та прізвище)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	4
ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	8
1.1 Огляд предметної області.....	8
1.2 Аналіз існуючих рішень	10
1.3 Визначення вимог системи.....	16
1.4 Постановка завдання.....	18
1.5 Моделювання предметної області.....	19
2 ПРОЄКТУВАННЯ СИСТЕМИ ПРИЙНЯТТЯ РІШЕНЬ	24
2.1 Загальна архітектура системи прийняття рішень	24
2.2 Логічна модель даних у вигляді ER-діаграми.....	28
2.3 Розробка структурної та принципової схем системи	30
2.4 Алгоритм збору, обробки та передачі даних.....	33
2.5 Опис протоколу взаємодії та інтерфейсів зв'язку	37
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ	39
3.1 Огляд середовищ розробки та інструментів програмування.....	39
3.2 Тестування системи	40
3.4 Впровадження системи.....	44
3.5 Аналіз точності вимірювання та стабільності передачі.....	46
4 ПИТАННЯ ЕНЕРГОЕФЕКТИВНОСТІ ТА НАДІЙНОСТІ.....	49
4.1 Оцінка енергоспоживання системи у різних режимах.....	49
4.2 Можливості масштабування та адаптації до різних сценаріїв використання	51
ВИСНОВКИ.....	53
СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55

ДОДАТОК А.....	58
----------------	----

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- АНР (Analytic Hierarchy Process) — метод аналітичної ієрархії для багатокритеріального прийняття рішень шляхом парних порівнянь альтернатив.
- API (Application Programming Interface) — програмний інтерфейс взаємодії між програмними модулями або зовнішніми інформаційними системами.
- DTO (Data Transfer Object) — структура даних, що використовується для передачі інформації між модулями системи.
- ELECTRE (Elimination and Choice Expressing Reality) — метод багатокритеріального аналізу на основі відношень переваг між альтернативами.
- ORM (Object-Relational Mapping) — об’єктно-реляційне відображення для взаємодії із системою керування базами даних.
- REST (Representational State Transfer) — архітектурний стиль побудови веб-сервісів для обміну інформацією між компонентами системи.
- SaaS (Software as a Service) — модель хмарного надання програмного забезпечення як сервісу.
- TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) — метод близькості до ідеального рішення для багатокритеріального ранжування альтернатив.
- OAuth 2.0 — протокол авторизації для захищеного керування сесіями доступу до ресурсів.
- SSL/TLS (Secure Sockets Layer / Transport Layer Security) — протоколи шифрування для забезпечення захищеного обміну інформацією в мережі.
- SQL (Structured Query Language) — мова структурованих запитів для роботи з реляційними базами даних.

- JSON (JavaScript Object Notation) — формат обміну даними між компонентами системи.
- XML (Extensible Markup Language) — стандарт розмітки для передачі структурованих даних у системах обміну інформацією.
- PK (Primary Key) — первинний ключ у базі даних, що забезпечує унікальність записів таблиці.
- FK (Foreign Key) — зовнішній ключ у базі даних, що реалізує зв'язок між таблицями.
- API Gateway — модуль проміжного рівня для маршрутизації запитів до внутрішніх та зовнішніх API-сервісів.
- Business Logic Module — модуль програмної системи, який реалізує основні правила обробки та аналізу вхідних даних.
- Criteria Manager — підсистема для управління критеріями оцінювання в системі прийняття рішень.
- Evaluation Module — модуль оцінювання альтернатив за визначеними критеріями.
- Normalization — процес приведення вхідних даних до уніфікованої шкали для забезпечення їх порівнюваності.
- Aggregation Module — підсистема обчислення інтегральних оцінок за результатами нормалізації та вагових коефіцієнтів.
- Reporting Module — модуль формування звітних документів за результатами аналізу.
- Decision Session — сесія прийняття рішень, що об'єднує конкретний набір введених даних, критеріїв та результатів обробки.
- Cloud Provider API — зовнішні інтерфейси хмарних постачальників, які надають доступ до актуальних параметрів сервісів.
- Uptime — показник безперервної доступності сервісу протягом певного періоду часу.

ВСТУП

Сучасний розвиток інформаційних технологій зумовлює активне впровадження хмарних обчислень у різні сфери діяльності організацій. Різноманітність хмарних сервісів, їх провайдерів та сервісних моделей потребує обґрунтованого вибору оптимальних рішень, що максимально відповідають потребам конкретної організації з урахуванням економічних, технічних, функціональних і безпекових критеріїв [1]. Вибір відповідної конфігурації хмарних сервісів є складним завданням, оскільки потребує врахування значної кількості змінних параметрів та ризиків. Традиційні підходи, засновані на експертних оцінках, є недостатньо ефективними у складних динамічних умовах ринку, що обумовлює актуальність автоматизованих систем підтримки прийняття рішень у цій сфері.

Системи підтримки прийняття рішень дозволяють забезпечити структурований аналіз доступних альтернатив, використовуючи математичні моделі багатокритеріального аналізу. Застосування таких методів дає змогу формалізувати процес порівняння варіантів, враховувати численні параметри та приймати обґрунтовані управлінські рішення. Водночас розробка подібних систем супроводжується низкою технічних труднощів, зокрема, щодо забезпечення коректної обробки вхідних даних, оптимального вибору методів оцінювання та побудови гнучкої архітектури, здатної адаптуватися до специфічних вимог користувача [2].

Метою даної роботи є розробка системи підтримки прийняття рішень при виборі хмарних сервісів для організацій, що забезпечує комплексну багатокритеріальну оцінку альтернатив з урахуванням актуальних вимог користувачів та особливостей ринку.

Для досягнення поставленої мети передбачено виконати такі **завдання**:

- проаналізувати сучасний стан та існуючі методи вибору хмарних сервісів;

- обґрунтувати вибір алгоритмів багатокритеріальної оцінки;
- розробити архітектуру системи підтримки прийняття рішень;
- реалізувати алгоритми збору, обробки та аналізу даних;
- створити прототип системи та забезпечити його тестування;
- провести аналіз ефективності розробленої системи.

Об'єктом дослідження є процес прийняття управлінських рішень щодо вибору хмарних сервісів для організацій.

Предметом дослідження є методи побудови систем підтримки прийняття рішень із використанням багатокритеріальних моделей у сфері хмарних обчислень.

Практична значущість роботи полягає у створенні адаптивного інструменту для автоматизованого вибору хмарних сервісів, що забезпечує формалізований підхід до прийняття рішень та може бути використаний у різних галузях діяльності організацій.

Структура кваліфікаційної роботи включає вступ, три основні розділи, висновки, список використаних джерел та додатки. У першому розділі представлено аналітичний огляд предметної області. У другому розділі здійснено розробку архітектури та алгоритмів функціонування системи. У третьому розділі наведено результати реалізації прототипу та експериментальних досліджень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Огляд предметної області

Сучасний розвиток інформаційних технологій зумовив стрімке зростання використання хмарних обчислень як ключового елемента інформаційної інфраструктури організацій. Хмарні сервіси забезпечують широкий спектр функціональних можливостей, включаючи зберігання даних, обробку інформації, запуск обчислювальних задач, організацію взаємодії та управління інформаційними потоками з різними рівнями гнучкості, масштабованості та вартості [3]. Це дозволяє підприємствам оптимізувати власні ресурси, зменшити витрати на підтримку локальної ІТ-інфраструктури, забезпечити стійкість до збоїв та оперативно адаптуватися до змін в ринкових умовах.

Основними моделями надання хмарних послуг є інфраструктура як послуга (IaaS), платформа як послуга (PaaS) та програмне забезпечення як послуга (SaaS). Модель IaaS дозволяє орендувати обчислювальні ресурси, сервери, сховища та мережеві компоненти. У свою чергу, PaaS надає програмні платформи для розробки та розгортання додатків, тоді як SaaS забезпечує доступ до готових програмних рішень через мережу без необхідності локальної інсталяції [4].

Процес вибору хмарних сервісів є складною задачею, що передбачає аналіз великої кількості параметрів. До основних вхідних даних, які необхідно враховувати при формуванні системи підтримки прийняття рішень, належать технічні, економічні та організаційні характеристики сервісів, що аналізуються. На рис. 1.1 узагальнено ключові групи параметрів, які слугують вхідною інформацією для багатокритеріальної оцінки.

У табл. 1.1 наведено основні групи вхідних даних для аналізу хмарних сервісів.

Таблиця 1.1

Основні групи вхідних даних для оцінки хмарних сервісів

Група параметрів	Основні характеристики
Технічні параметри	Обчислювальна потужність, об'єм сховищ, пропускна здатність каналів, надійність інфраструктури
Економічні параметри	Вартість оренди ресурсів, ліцензійні платежі, витрати на підтримку, гнучкість тарифних планів
Безпекові параметри	Захист даних, сертифікація провайдера, політика конфіденційності, відповідність стандартам
Юридичні аспекти	Територіальне розташування центрів обробки даних, відповідність локальному законодавству
Сервісна підтримка	Доступність технічної підтримки, швидкість реагування, рівень SLA
Функціональні можливості	Наявність додаткових сервісів: аналітики, резервного копіювання, автоматизації

Різноманітність даних, що аналізуються, потребує їх попередньої нормалізації та уніфікації для коректної роботи алгоритмів багатокритеріальної обробки. Частина параметрів подається у вигляді кількісних значень, інша частина — у вигляді якісних оцінок або експертних шкал, які необхідно перевести у числовий формат для подальшої обробки. Джерелами вхідних даних можуть слугувати офіційна документація хмарних провайдерів, публічні API сервісів, технічні паспорти інфраструктури, стандартизовані рейтинги, а також результати опитувань внутрішніх IT-підрозділів організації.

Важливим аспектом є врахування динамічності вхідних параметрів. Наприклад, вартість послуг, пропускна здатність мережі, або політики доступу до даних можуть змінюватися в процесі експлуатації сервісу. Це потребує впровадження періодичного оновлення інформації в системі для забезпечення актуальності результатів прийняття рішень.

Узагальнюючи, слід зазначити, що ефективна система підтримки прийняття рішень повинна забезпечити інтеграцію різних джерел вхідних даних, їхню попередню обробку, формування уніфікованої бази параметрів для

багатокритеріальної оцінки, а також підтримку механізмів гнучкого налаштування вагових коефіцієнтів залежно від актуальних потреб організації.

1.2 Аналіз існуючих рішень

У процесі вибору хмарних сервісів сьогодні існує низка спеціалізованих програмних рішень та онлайн-платформ, що забезпечують аналітичну підтримку користувачів шляхом порівняння характеристик, вартості та ефективності використання хмарних ресурсів. Розглянемо кілька актуальних прикладів таких систем.

Одним із відомих сервісів є CloudHarmony, який спеціалізувався на наданні об'єктивної інформації щодо продуктивності хмарних провайдерів. Система виконувала незалежне тестування продуктивності обчислювальних, мережових та сховищевих ресурсів у різних географічних регіонах. Основна функціональність CloudHarmony полягала у зборі великого масиву тестових даних для забезпечення прозорого вибору хмарних послуг відповідно до реальних показників продуктивності. Однак варто зазначити, що з травня 2023 року цей інструмент припинив свою роботу, а оновлення даних було зупинене ще раніше, у лютому 2023 року. Це обмежує його застосування для актуальних порівнянь у сучасних проектах. Інтерфейс та функціональні можливості сервісу наведено на рисунку 1.1.

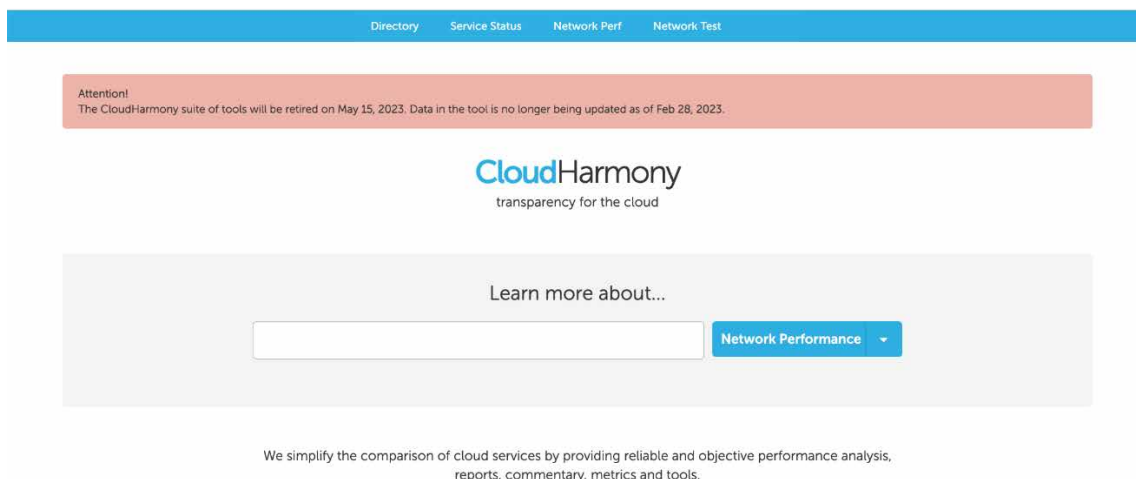


Рисунок 1.1 – Інтерфейс платформи CloudHarmony

Іншим дієвим прикладом є система Clouddorado, яка орієнтується переважно на оцінку економічних характеристик хмарних серверів. Даний інструмент дозволяє користувачу задавати параметри конфігурації віртуальної машини, такі як обсяг оперативної пам'яті, обсяг дискового простору, продуктивність процесора та операційну систему. На основі цих даних система автоматично здійснює пошук серед наявних хмарних провайдерів, формуючи порівняльну таблицю із зазначенням вартості оренди обраних ресурсів. Clouddorado дозволяє зосередити увагу саме на економічних аспектах хмарної інфраструктури, що є важливим для організацій з обмеженими фінансовими ресурсами. Роботу інтерфейсу сервісу представлено на рисунку 1.2.

Cloudorado Cloud Server Cloud Hosting Cloud Providers FAQ Contact Blog

Cloud Server Comparison

Calculate cloud server price and make your custom cloud hosting cost comparison. Use sliders to set your cloud server requirements and narrow cloud hosting offers with filters on the left, result with cloud server prices will update instantly. Cloud servers have typically the biggest contribution in the cloud hosting infrastructure costs so it is important to make thorough analysis which cloud sever provider is really the best for you.

Location:
Any location

RAM: 512 768 1G 1.5G 2G 4G 8G 16G 32G 64G

Storage: 1GB 2 5 10 20 50 100 200 500 1TB

CPU Power: Any 1x 2x 3x 4x 5x 6x 7x 8x
eq. Xeon E5520

OS: Linux Windows [More options](#)

Cloud Features & Management

- Hourly Pay-As-You-Go
- One Account For All Locations
- Web Interface
- Mobile Interface
- Terminal access
- API
- AWS-compatible API
- OpenStack-compatible API
- Auto-scaling
- Vertical scaling without reboot
- Image from cloud server

[show more](#)

26 cloud server providers found

Cloud Provider	Cloud Server Summary	Price
	1.0 GB RAM / 1 VCPU show details	\$35 Go to Provider
	SSD 1 GB RAM / 2.4 GHz VCPU (Small) show details	\$10 Go to Provider

Рисунок 1.2 – Інтерфейс платформи Cloudorado

Більш комплексною системою виступає RightScale Optima, яка забезпечує багатовимірний аналіз витрат на хмарну інфраструктуру. Система дозволяє проводити детальну аналітику витрат за категоріями, прогнозувати зміни бюджетних показників, аналізувати історичну динаміку використання ресурсів та моделювати сценарії оптимізації. Особливістю RightScale Optima є підтримка багатоплатформеного аналізу, що дозволяє одночасно оцінювати витрати для різних хмарних провайдерів, інтегруючи дані у єдину фінансову модель. Загальний вигляд аналітичної панелі RightScale Optima представлено на рисунку 1.3.

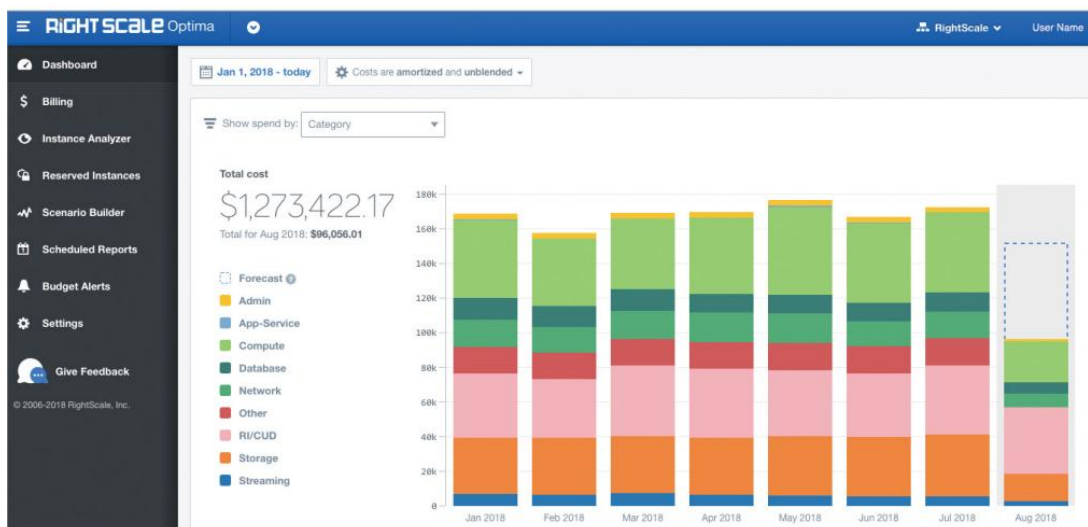


Рисунок 1.3 – Інтерфейс системи RightScale Optima

Інноваційний підхід демонструє система CAST AI, яка орієнтована на автоматизовану оптимізацію витрат при використанні хмарних середовищ. Дана платформа інтегрується з існуючими кластерами користувача та проводить автоматичний аналіз конфігурацій, виявляє неефективні налаштування і пропонує рекомендації для зниження витрат. Однією з ключових функцій є механізм autoscaling, що дозволяє динамічно коригувати ресурси у відповідь на зміну навантаження, зменшуючи надлишкові витрати. Інтерфейс системи CAST AI зображено на рисунку 1.4.

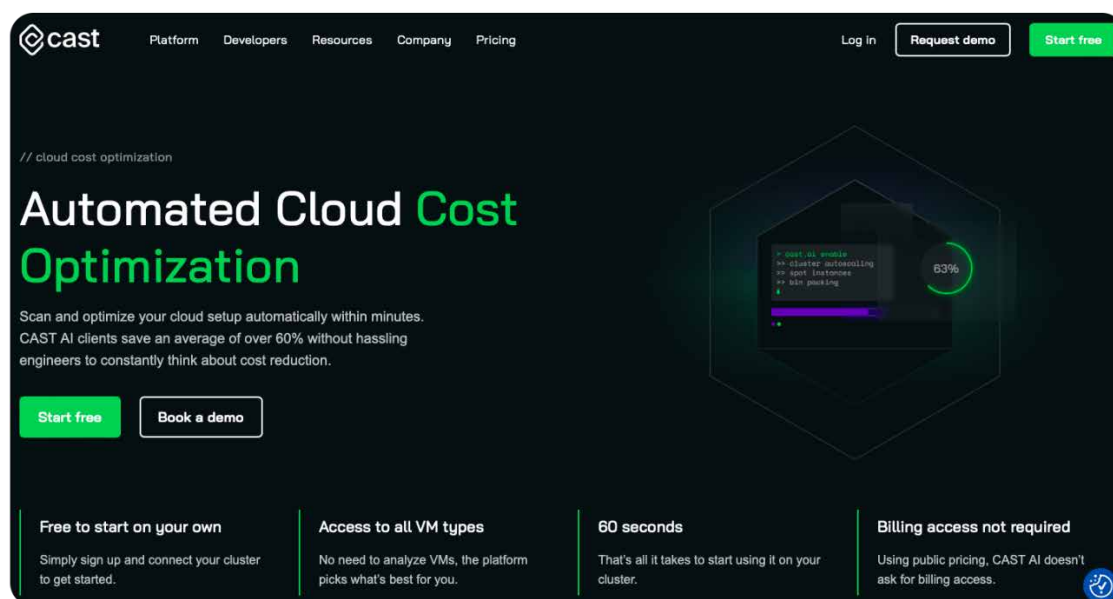


Рисунок 1.4 – Інтерфейс системи CAST AI

Ще одним поширеним інструментом є Flexera One, який дозволяє здійснювати централізоване управління хмарними витратами, відстеження вартості використання сервісів та оптимізацію фінансових потоків у складних мультихмарних середовищах. Система підтримує інтеграцію з основними хмарними провайдерами, формує розширені звітні панелі, забезпечує прозорість розподілу витрат та дозволяє управляти ліцензіями та обліковими записами. Вигляд основної аналітичної панелі системи Flexera One наведено на рисунку 1.5

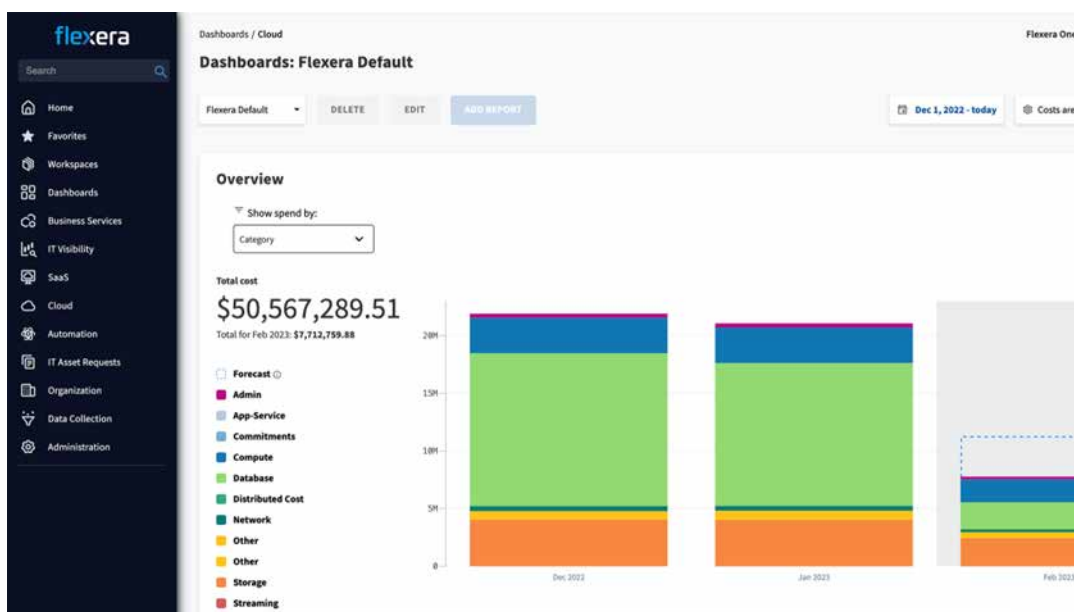


Рисунок 1.5 – Інтерфейс системи Flexera One

Проведений аналіз свідчить про наявність широкого спектру інструментів для підтримки прийняття рішень у сфері вибору хмарних сервісів. Водночас кожне з розглянутих рішень має власну спеціалізацію, обмеження та цільову аудиторію, що обумовлює доцільність розробки адаптивних універсальних систем, здатних інтегрувати переваги існуючих підходів та враховувати індивідуальні потреби організацій.

З урахуванням проведеного аналізу існуючих рішень можна зробити висновок, що сучасні системи підтримки прийняття рішень при виборі хмарних сервісів характеризуються значною варіативністю за функціональними можливостями, методами збору даних, рівнем автоматизації та типами користувацьких завдань. Кожен із розглянутих інструментів орієнтований на власну цільову аудиторію та вирішує окремі аспекти проблеми. Водночас жодне

з існуючих рішень не забезпечує комплексної багатокритеріальної підтримки з урахуванням повного спектру технічних, економічних, організаційних та правових факторів.

Платформа CloudHarmony була зосереджена переважно на забезпеченні прозорості продуктивності та порівнянні ключових технічних показників. Система Cloudorado акцентувала увагу на порівнянні економічних параметрів оренди хмарних ресурсів. RightScale Optima дозволяла комплексно аналізувати витрати з детальним фінансовим прогнозуванням, орієнтуючись на великі корпоративні середовища. Платформа CAST AI надавала механізми автоматизованої оптимізації конфігурацій з динамічним масштабуванням для зменшення затрат. Flexera One забезпечувала централізоване управління витратами в мультихмарних інфраструктурах із підтримкою широкого спектру інтеграцій.

Зведене порівняння основних характеристик існуючих рішень наведено у таблиці 1.2.

Таблиця 1.2

Порівняння існуючих систем підтримки прийняття рішень при виборі хмарних сервісів

Система	Основна спеціалізація	Методи аналізу	Рівень автоматизації	Орієнтація на аудиторію
	Оцінка продуктивності	Тестові бенчмарки	Низький	Технічні експерти
	Порівняння вартості	Калькулятор параметрів	Середній	Малі та середні підприємства
	Фінансовий аналіз витрат	Прогнозування та симуляція	Середній	Великі корпорації
	Автоматична оптимізація конфігурацій	Машинне навчання,	Високий	DevOps та інженери хмарних платформ
	Централізоване управління витратами	Моніторинг витрат, інтеграції	Високий	Мультихмарні корпоративні середовища

Існуючі інструменти демонструють високу спеціалізацію, що створює передумови для розробки універсальної системи, яка б поєднувала можливості попередньої обробки вхідних даних, багатокритеріального аналізу та автоматизації процесу прийняття рішень з урахуванням актуальних потреб конкретної організації.

1.3 Визначення вимог системи

Формування вимог до системи підтримки прийняття рішень при виборі хмарних сервісів ґрунтується на особливостях предметної області, технічних обмеженнях, актуальних практиках застосування хмарних технологій, а також специфіці управлінських процесів в організаціях. Комплексний аналіз вимог дозволяє забезпечити повноту охоплення критеріїв, необхідних для прийняття обґрунтованих рішень, та формує основу для подальшого проєктування архітектури системи [7].

Загальні функціональні вимоги визначають базовий спектр задач, які повинна виконувати система. До таких вимог належать збір вхідних даних про параметри хмарних сервісів, забезпечення багатокритеріального аналізу альтернатив, підтримка алгоритмів нормалізації даних, формування зведених звітів, можливість порівняння декількох провайдерів одночасно, а також зручність взаємодії з кінцевим користувачем. Узагальнені функціональні вимоги наведено у таблиці 1.3.

Таблиця 1.3

Основні функціональні вимоги до системи

Вимога	Опис
Збір даних	Автоматизоване або ручне отримання технічних, фінансових, безпекових та організаційних параметрів
Обробка даних	Нормалізація показників, перетворення якісних оцінок у кількісні
Багатокритеріальний аналіз	Реалізація одного або кількох методів прийняття рішень (АНР, TOPSIS тощо)

Продовження таблиці 1.3

Формування звітів	Генерація таблиць результатів, графічних діаграм, рейтингів альтернатив
Підтримка інтерактивності	Можливість зміни вагових коефіцієнтів користувачем
Масштабованість	Підтримка розширення системи на нові критерії та джерела даних

Окрім функціональних характеристик, система повинна відповідати низці нефункціональних вимог, що стосуються якості її роботи, стабільності, надійності та безпеки даних. До нефункціональних вимог відносяться вимоги до продуктивності, конфіденційності, адаптивності до змін параметрів аналізу та стійкості до збоїв у роботі зовнішніх джерел даних. Узагальнення нефункціональних вимог представлено у таблиці 1.4.

Таблиця 1.4

Основні нефункціональні вимоги до системи

Вимога	Опис
Продуктивність	Забезпечення обробки запитів у межах встановлених часових нормативів
Надійність	Стійка робота при неповних або застарілих даних
Конфіденційність	Захист персональних та конфіденційних організаційних даних
Масштабованість	Можливість інтеграції нових критеріїв, модулів та алгоритмів
Портативність	Незалежність від апаратних платформ та операційних систем
Доступність	Забезпечення безперервного доступу до системи в робочому режимі

Детальний опис вимог до вхідних даних відіграє ключову роль у коректності функціонування системи. На етапі збору інформації необхідно враховувати різноманітні формати вхідних параметрів. Вхідні дані можуть надходити з внутрішніх систем управління організації, публічних API хмарних провайдерів, відкритих технічних специфікацій, а також результатів опитувань експертів. Основні групи вхідних даних наведено у таблиці 1.5.

Таблиця 1.5

Групи вхідних даних для аналізу

Джерело	Параметри
Технічні характеристики	Обчислювальні ресурси, пропускна здатність мережі, типи зберігання
Фінансові показники	Вартість оренди, модель тарифікації, додаткові платежі
Параметри безпеки	Сертифікати відповідності, політики захисту даних, шифрування
Юридичні аспекти	Геолокація обробки даних, відповідність законодавству
Параметри підтримки	Рівень SLA, канали підтримки, регламенти обслуговування

Комплексний аналіз вимог дозволяє сформувати структуровану основу для розробки архітектури системи підтримки прийняття рішень, орієнтованої на інтеграцію різномірних джерел інформації, ефективно багатокритеріальне оцінювання альтернатив та підтримку актуальних потреб користувачів у сфері хмарних обчислень [8].

1.4 Постановка завдання

З урахуванням проведеного аналізу предметної області, існуючих рішень та сформованих вимог, формулюється загальна постановка задачі, яка визначає основні функціональні компоненти та логіку роботи розроблюваної системи підтримки прийняття рішень при виборі хмарних сервісів. Система повинна забезпечити комплексну обробку вхідних параметрів, здійснювати багатокритеріальну оцінку доступних альтернатив і формувати рекомендації для кінцевого користувача на основі інтегрованих оцінок.

Основною задачею є створення адаптивної моделі багатокритеріального аналізу, що дозволяє інтегрувати технічні, економічні, безпекові та організаційні параметри хмарних сервісів у єдину оціночну систему. Модель повинна підтримувати можливість коригування вагових коефіцієнтів, нормалізації вхідних даних, проведення обчислень інтегральних оцінок та ранжування альтернатив. Загальна структура постановки задачі наведена у таблиці 1.6.

Формалізація постановки задачі

Компонент	Опис
Вхідні дані	Сукупність параметрів хмарних сервісів: технічних, економічних, безпекових, юридичних та сервісних
Попередня обробка	Перетворення вхідних параметрів у нормалізовану шкалу, уніфікація одиниць виміру, обробка відсутніх значень
Визначення ваг	Призначення вагових коефіцієнтів для кожного критерію відповідно до їх пріоритетності
Обчислювальна модель	Застосування багатокритеріального методу для інтегрального оцінювання (наприклад, АНР, TOPSIS)
Формування результатів	Розрахунок ранжування альтернатив та генерація звітів з підсумковими показниками
Вивід даних	Представлення результатів користувачеві у табличному та графічному вигляді

Задача передбачає побудову універсальної моделі, яка дозволяє працювати з різними джерелами даних, динамічно адаптувати ваги критеріїв та забезпечувати коректність обчислень при наявності часткових чи неповних даних. Особлива увага приділяється створенню механізмів нормалізації, оскільки вхідні параметри можуть мати різні діапазони значень та різні одиниці виміру.

Система має бути орієнтована як на автоматичний режим роботи при наявності повних масивів даних з API, так і на режим експертної оцінки у випадку часткових або обмежених вхідних даних. Такий підхід забезпечує універсальність моделі та її придатність для застосування в організаціях з різним рівнем інформаційної забезпеченості [10].

1.5 Моделювання предметної області

Моделювання предметної області є важливим етапом формалізації функціональної структури системи підтримки прийняття рішень при виборі хмарних сервісів. Використання уніфікованих моделей дозволяє відобразити

основних учасників процесу, їхні ролі, взаємодії, етапи обробки даних та логіку виконання бізнес-процесів системи.

Початковим етапом моделювання є побудова діаграми варіантів використання, що дозволяє визначити основних акторів системи та описати їх взаємодію з ключовими функціональними компонентами. У системі виділено три основних групи користувачів: користувач, адміністратор та експерт. Користувач здійснює введення критеріїв та отримує рекомендації. Адміністратор запускає процес прийняття рішень та забезпечує актуальність бази даних. Експерт має додаткові повноваження з редагування критеріїв та роботи з історією прийнятих рішень. Структуру варіантів використання наведено на рисунку 1.6.

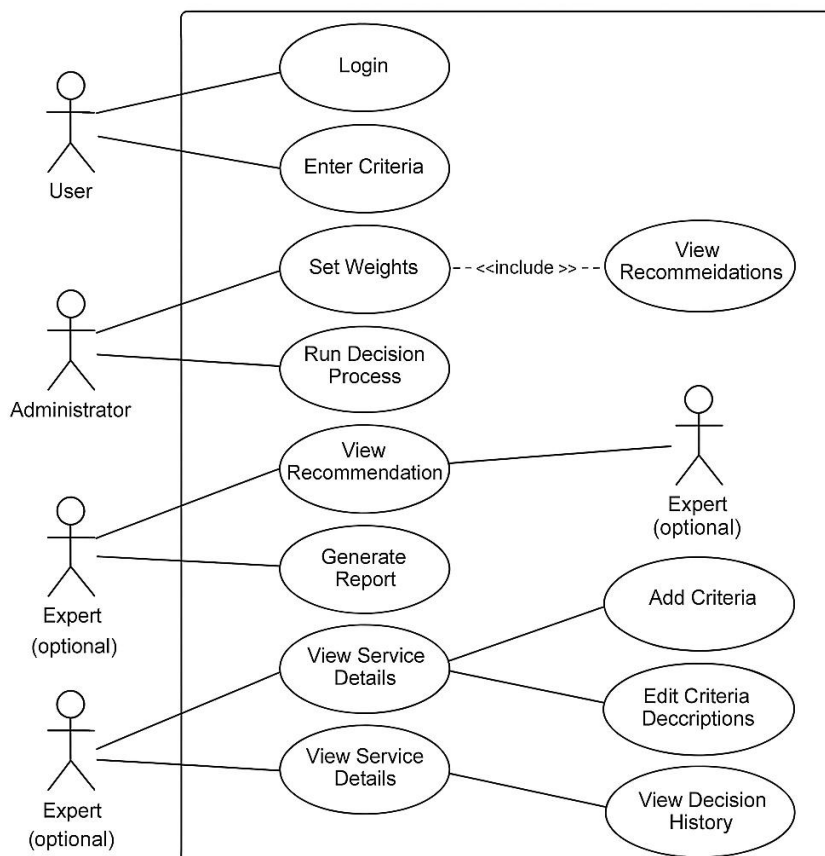


Рисунок 1.6 – Діаграма варіантів використання системи

Логіку виконання основного бізнес-процесу відображає діаграма діяльності. Процес починається з авторизації користувача та введення критеріїв. У разі необхідності користувач може коригувати вагові коефіцієнти для критеріїв. Далі здійснюється завантаження даних про хмарні сервіси,

нормалізація вхідних значень, виконання алгоритму прийняття рішень, ранжування альтернатив та відображення результатів. За необхідності формується додатковий аналітичний звіт. Повну послідовність операцій представлено на рисунку 1.7.

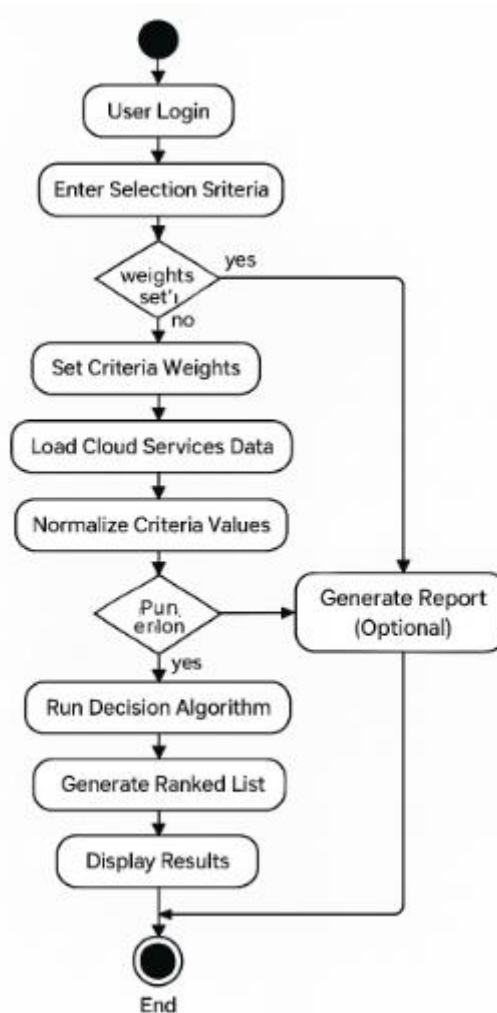


Рисунок 1.7 – Діаграма діяльності процесу прийняття рішень

Для деталізації механізмів взаємодії між компонентами системи сформовано діаграму послідовності. У ній відображено послідовний обмін повідомленнями між інтерфейсом користувача, бізнес-логікою, процесором алгоритмів прийняття рішень, агрегатором результатів, генератором звітів та шаром доступу до даних. Ця модель демонструє архітектурну організацію обробки запитів, що забезпечує розділення відповідальностей між модулями. Логіка взаємодії наведена на рисунку 1.8.

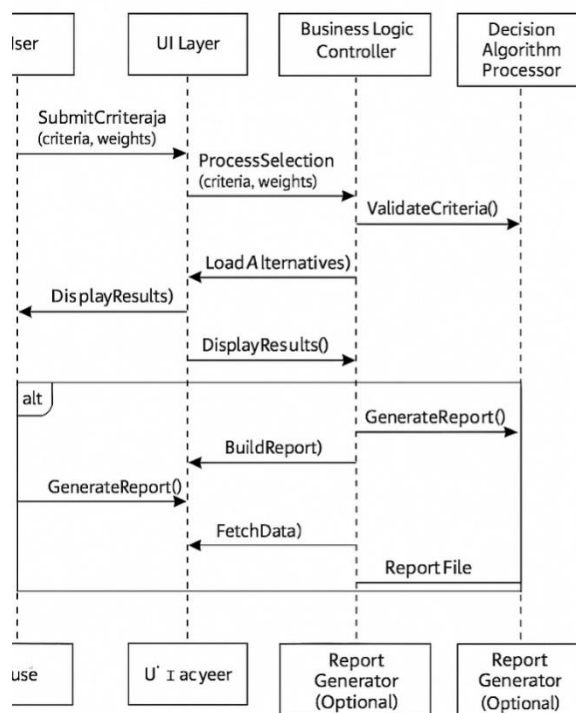


Рисунок 1.8 – Діаграма послідовності обробки запитів у системі

Функціональне наповнення системи деталізується за допомогою узагальненої таблиці основних функціональних компонентів, що наведена у таблиці 1.7.

Таблиця 1.7

Основні функціональні компоненти системи

Компонент	Основні функції
Інтерфейс користувача	Введення критеріїв, коригування ваг, запуск обчислень, перегляд результатів
Модуль збору даних	Отримання технічних та фінансових параметрів сервісів
Модуль нормалізації	Перетворення показників у єдину шкалу
Обчислювальний модуль	Виконання алгоритмів багатокритеріального аналізу
Модуль агрегування	Формування інтегральних оцінок та ранжування альтернатив
Модуль звітності	Формування аналітичних звітів для користувача
Адміністративний модуль	Управління базою критеріїв та історією рішень

Важливою складовою є також формалізація потоків вхідних та вихідних даних. Джерела, типи та способи використання даних систематизовано у таблиці 1.8.

Таблиця 1.8

Потоки даних у системі

Потік даних	Джерело	Використання
Критерії та ваги	Користувач, експерт	Формування оціночної моделі
Параметри сервісів	База даних, API	Оцінка технічних та фінансових показників
Результати обчислень	Обчислювальний модуль	Ранжування альтернатив
Звіти	Модуль звітності	Надання аналітичної інформації користувачу

Побудовані моделі предметної області дозволяють сформулювати цілісну концептуальну основу для подальшого проектування архітектури системи підтримки прийняття рішень при виборі хмарних сервісів [11].

2 ПРОЄКТУВАННЯ СИСТЕМИ ПРИЙНЯТТЯ РІШЕНЬ

2.1 Загальна архітектура системи прийняття рішень

Архітектура системи підтримки прийняття рішень при виборі хмарних сервісів будується на основі модульної багаторівневої структури, що забезпечує розділення функціональних відповідальностей, гнучкість масштабування та спрощення подальшої підтримки і розвитку системи. Основною метою побудови архітектури є організація взаємодії між ключовими компонентами, що реалізують збір, обробку, аналіз та візуалізацію даних, необхідних для прийняття управлінських рішень.

Загальна логічна структура програмної системи визначає основні компоненти, які реалізують ключові функції системи: введення даних, бізнес-логіку, доступ до даних, зовнішню інтеграцію, формування звітів та забезпечення безпеки доступу. Структуру основних компонентів програмного забезпечення наведено на рисунку 2.1.

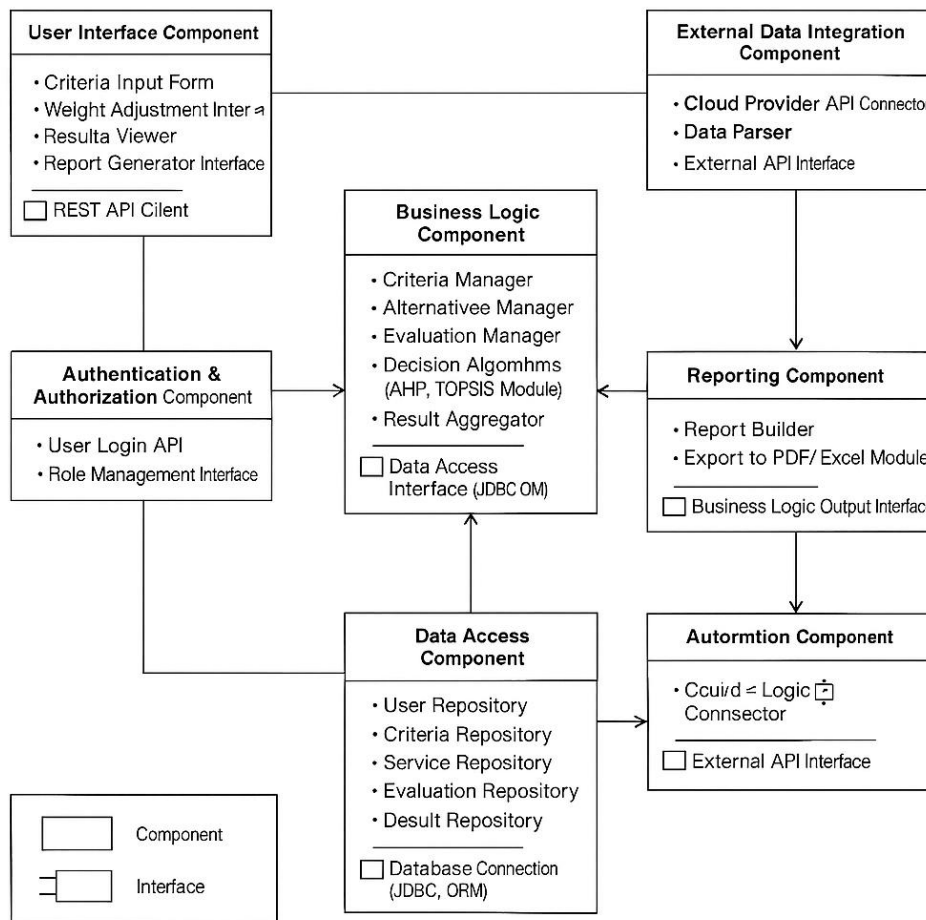


Рисунок 2.1 – Компонентна діаграма архітектури системи

Компонент «User Interface» забезпечує взаємодію користувача з системою. Він реалізує введення критеріїв, налаштування вагових коефіцієнтів, запуск процесу обробки даних, перегляд отриманих рекомендацій та генерацію звітів. Компонент «Authentication & Authorization» забезпечує автентифікацію користувачів та управління правами доступу до функціональних модулів системи.

Центральним елементом є компонент «Business Logic», який включає в себе управління критеріями, альтернативами, оцінками, обробку даних та виконання алгоритмів прийняття рішень. Взаємодія з базою даних здійснюється через компонент «Data Access», що організовує роботу із сховищами критеріїв, сервісів, оцінок, рішень та результатів. Компонент «External Data Integration» відповідає за отримання актуальних даних від зовнішніх провайдерів.

Завершальним елементом є «Reporting Component», що генерує фінальні звіти на основі розрахованих результатів.

Архітектурні взаємозв'язки системи із зовнішніми користувачами та інформаційними потоками деталізовано на рисунку 2.2.

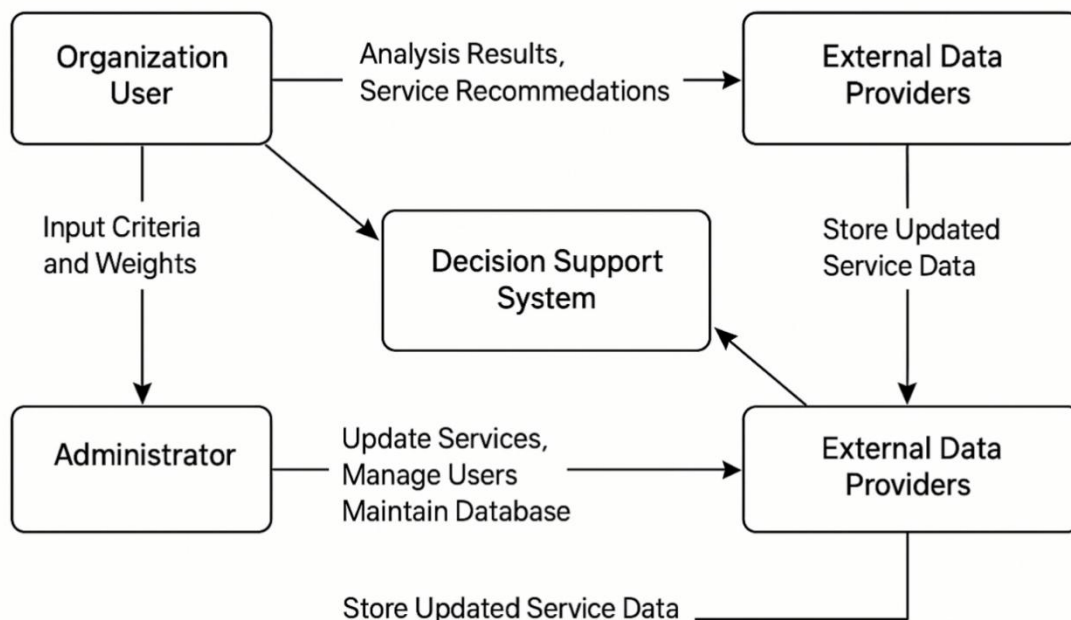


Рисунок 2.2 – Контекстна діаграма системи підтримки прийняття рішень

Користувачі взаємодіють із системою шляхом введення критеріїв оцінювання та вагових коефіцієнтів. Адміністратор забезпечує оновлення баз даних сервісів, управління обліковими записами користувачів та обслуговування системи. Система взаємодіє із зовнішніми джерелами даних для отримання оновленої інформації про параметри хмарних сервісів та зберігає отримані дані у власних сховищах.

Основна логіка внутрішньої роботи програмного забезпечення деталізується за допомогою діаграми потоків даних, що представлена на рисунку 2.3.

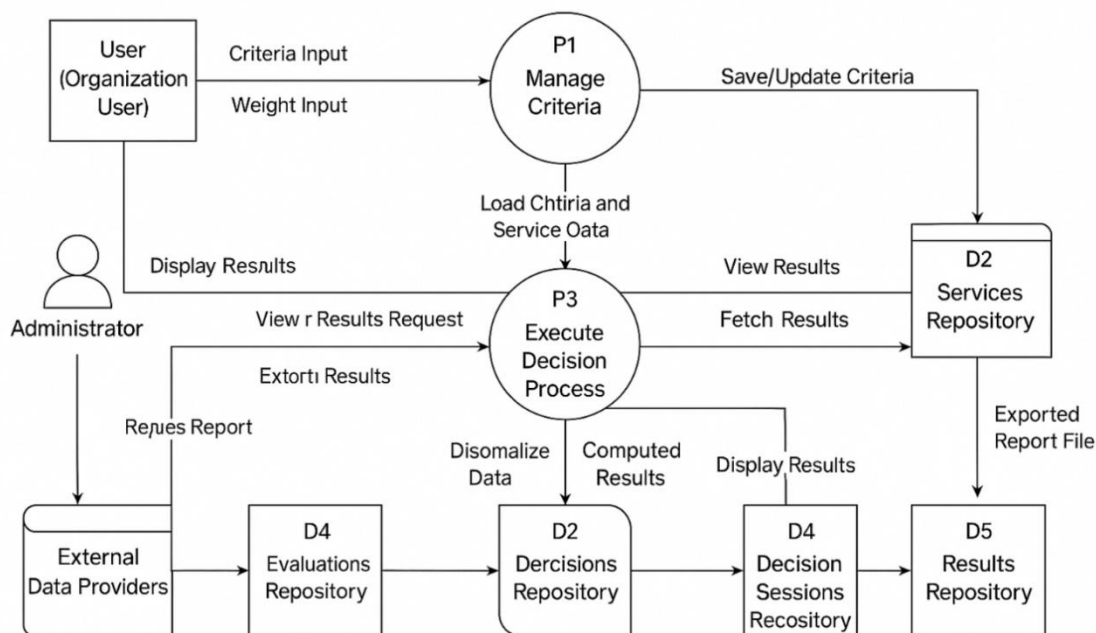


Рисунок 2.3 – Діаграма потоків даних системи

Дані, які вводяться користувачем, надходять у модуль керування критеріями. Після цього система завантажує актуальну інформацію про сервіси із відповідних репозиторіїв. Модуль виконання обчислювального процесу прийняття рішень проводить нормалізацію даних та обчислення інтегральних оцінок. Отримані результати зберігаються у репозиторії рішень та можуть бути виведені у вигляді звітів. Додатково підтримується збереження історії прийнятих рішень та оцінок.

Основні внутрішні модулі системи формалізовано у таблиці 2.1.

Таблиця 2.1

Внутрішні модулі архітектури програмного забезпечення

Модуль	Функціональні завдання
Менеджер критеріїв	Управління переліком критеріїв, редагування параметрів
Менеджер альтернатив	Формування списку доступних хмарних сервісів
Менеджер оцінок	Збір та обробка параметрів для кожного сервісу
Обчислювальний процесор	Виконання багатокритеріальних алгоритмів
Модуль нормалізації	Уніфікація значень критеріїв у спільну шкалу

Продовження таблиці 2.1

Агрегатор результатів	Розрахунок інтегральних оцінок та ранжування
Генератор звітів	Формування звітних документів на основі результатів
Модуль безпеки	Керування автентифікацією та доступом до системи
Інтерфейс даних	Робота із внутрішніми та зовнішніми сховищами

Побудована архітектурна модель забезпечує повноцінне розділення обов'язків між компонентами системи, дозволяє реалізувати ефективну обробку інформації, забезпечити гнучкість та масштабованість, а також сформувати технічну основу для подальшої деталізації алгоритмів прийняття рішень [12].

2.2 Логічна модель даних у вигляді ER-діаграми

Структурна організація даних у системі підтримки прийняття рішень при виборі хмарних сервісів базується на побудові логічної моделі даних, що формалізує взаємозв'язки між ключовими інформаційними сутностями. Логічна модель відображає повний життєвий цикл даних: від початкового введення критеріїв і оцінок до збереження результатів аналізу та підтримки історичних сесій прийняття рішень.

У моделі виділено кілька основних сутностей, що формують ядро системи. Сутність користувача (USER) зберігає облікову інформацію про зареєстрованих користувачів, включаючи їхній ідентифікатор, ім'я, електронну пошту, захищений пароль і роль доступу. Ролі визначають можливість доступу до адміністративних функцій або редагування критеріїв.

Ключовою складовою системи є сутність критеріїв (CRITERION), що містить опис кожного критерію оцінювання, його вагу та одиницю вимірювання. Це забезпечує універсальність формування моделей оцінювання для різних категорій хмарних сервісів.

Кожен акт прийняття рішення оформлюється у вигляді окремої сесії (DECISION_SESSION), що дозволяє зберігати повну історію аналітичних процесів. Для кожної сесії додатково фіксуються вагові коефіцієнти критеріїв у

таблиці `SESSION_CRITERIA`, що забезпечує збереження варіативних конфігурацій ваг для різних аналізів.

Для відображення конкретних оцінок критеріїв щодо альтернативних хмарних сервісів використовується сутність `EVALUATION`, яка фіксує значення оцінок для кожного критерію в межах заданої сесії. Результати багатокритеріального аналізу обчислюються та зберігаються у таблиці `RESULT`, що містить інтегральну оцінку кожної альтернативи та її позицію в ранжованому списку.

Зведену структуру логічної моделі даних представлено на рисунку 2.4.

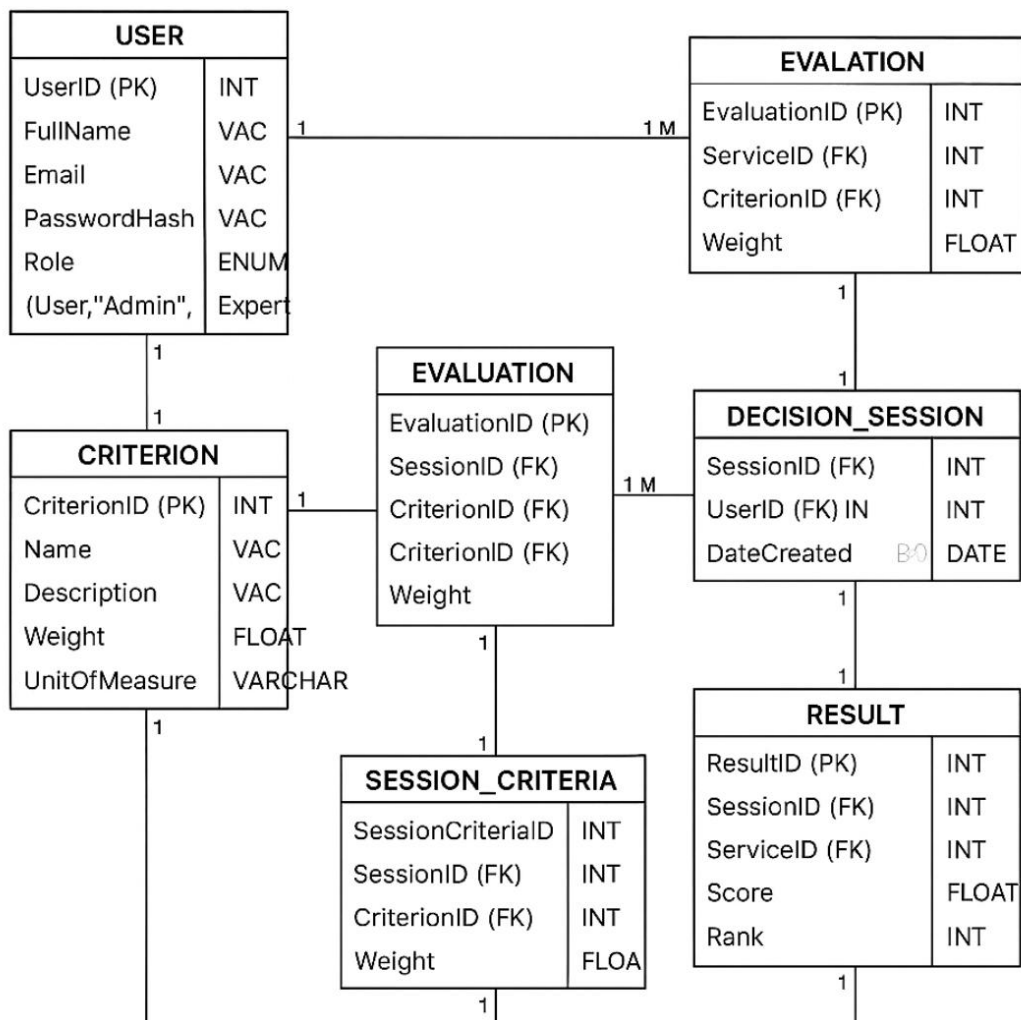


Рисунок 2.4 – Логічна ER-модель даних системи підтримки прийняття рішень

Кожна із зазначених сутностей містить набір основних атрибутів, що забезпечують повноцінне відображення інформаційних об'єктів системи. Основні атрибути сутностей наведено у таблиці 2.2.

Таблиця 2.2

Основні атрибути сутностей логічної моделі даних

Сутність	Атрибути

Побудована логічна структура забезпечує повну підтримку ключових функціональних процесів системи: збереження критеріїв та їх ваг, фіксацію оцінок сервісів, формування результатів багатокритеріальної обробки, підтримку історії прийняття рішень та генерацію звітів. Відповідна структура даних дозволяє організувати ефективний та надійний процес роботи системи як з технічної, так і з прикладної точок зору [13].

2.3 Розробка структурної та принципової схем системи

Розробка структурної та принципової схем системи підтримки прийняття рішень при виборі хмарних сервісів базується на попередньо визначених функціональних вимогах та архітектурних рішеннях. Моделі, що наведено нижче, забезпечують формалізацію процесів збору, обробки, аналізу даних та відображення результатів для кінцевого користувача.

Ключовим етапом формування архітектурної логіки системи є визначення послідовних фаз обробки інформації, починаючи від завантаження початкових даних до виведення остаточних результатів. Узагальнену модель основних етапів обробки даних наведено на рисунку 2.5.

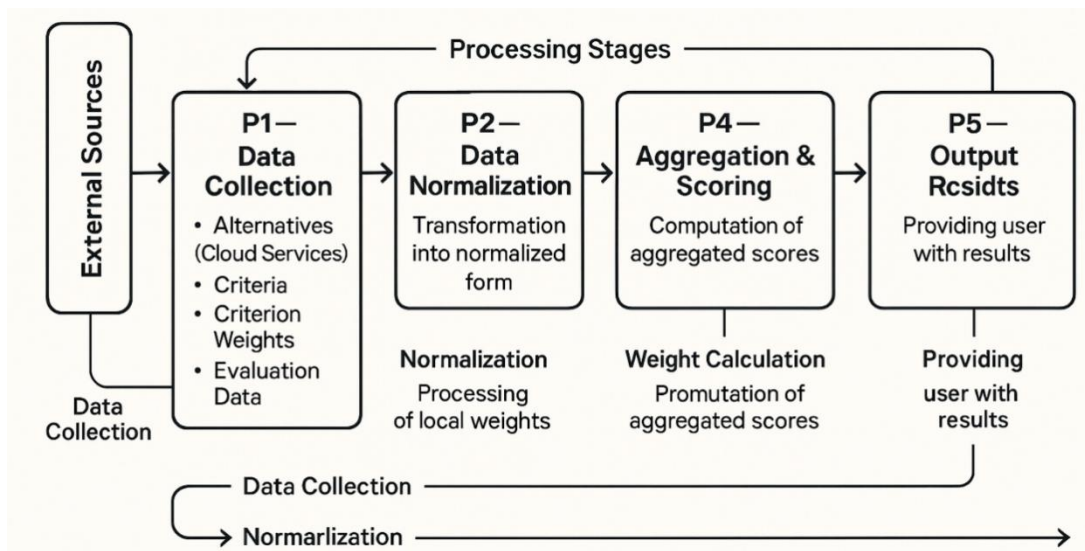


Рисунок 2.5 – Принципова схема етапів обробки даних

Згідно із зображеною схемою, обробка даних виконується в п'ять основних послідовних фаз. На першій фазі (P1) здійснюється збір даних, який охоплює отримання альтернатив (хмарних сервісів), критеріїв, їх вагових коефіцієнтів та оцінок. Джерела формування вхідних даних включають як внутрішні репозиторії системи, так і зовнішні провайдери через API.

Друга фаза (P2) відповідає за нормалізацію даних. У цій фазі відбувається трансформація різнорідних числових параметрів у уніфіковану нормалізовану форму, що дозволяє забезпечити коректну математичну обробку критеріїв, представлених у різних шкалах.

Третя фаза (P3) включає обчислення вагових коефіцієнтів критеріїв, а також перетворення локальних ваг у глобальні з урахуванням особливостей обраного алгоритму.

Четверта фаза (P4) забезпечує агрегацію нормалізованих оцінок та обчислення інтегральних показників ефективності альтернатив. Підсумковий етап (P5) формує остаточні результати та передає їх користувачеві у вигляді рейтингового списку або деталізованих звітів.

Для детального відображення внутрішніх функціональних взаємодій системи розроблено структурну модель основних логічних модулів, наведену на рисунку 2.6.

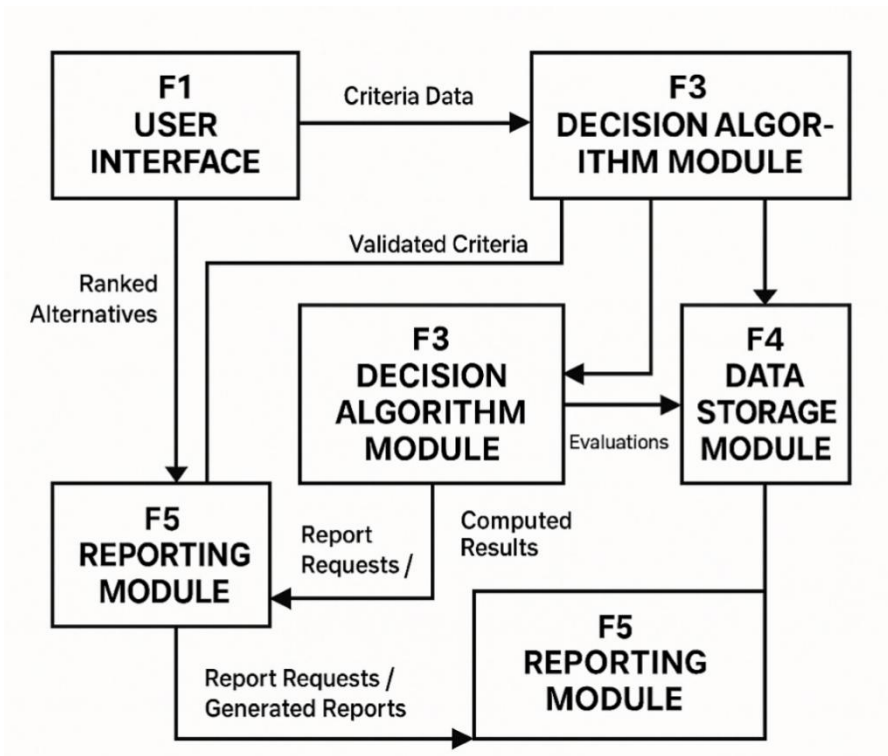


Рисунок 2.6 – Структурна схема логічних модулів системи

Модуль F1 (User Interface) забезпечує обробку введення критеріїв, отримання даних від користувача, ініціює запуск обчислювального процесу та отримує результати ранжування. Усі критерії передаються для подальшої обробки в модуль алгоритмів прийняття рішень.

Модуль F3 (Decision Algorithm Module) виконує перевірку валідності отриманих критеріїв, а також реалізує логіку багатокритеріальної обробки. Усі необхідні оцінки модуль отримує із модуля збереження даних F4 (Data Storage Module), що є єдиною точкою зберігання поточних і історичних даних оцінок, критеріїв, альтернатив та ваг.

Після обчислення модуль F5 (Reporting Module) здійснює формування запитів до збережених результатів і створює звітні документи для подальшого аналізу. Генерація звітів підтримує можливість експорту результатів для зовнішнього аналізу.

Інформаційна взаємодія модулів підтримується чіткою ізольованою передачею валідованих даних між складовими системи, що забезпечує стабільність роботи у випадку часткових відмов окремих модулів.

Додатково сформовано узагальнену таблицю основних функціональних підсистем системи, яка наведена у таблиці 2.3.

Таблиця 2.3

Основні функціональні підсистеми системи

Підсистема	Опис функціональності
Інтерфейс користувача	Введення критеріїв, запуск обчислень, перегляд результатів
Підсистема валідації	Перевірка коректності введених критеріїв і ваг
Модуль збору даних	Інтеграція з репозиторіями даних та зовнішніми джерелами
Алгоритмічний процесор	Реалізація методів АНР, TOPSIS, агрегація оцінок
Підсистема нормалізації	Перетворення даних у нормалізовану шкалу обробки
Сховище даних	Збереження критеріїв, оцінок, результатів і сесій
Модуль звітності	Формування детальних звітів на основі результатів

Запропоновані структурна та принципова моделі програмної системи забезпечують повний замкнутий цикл обробки вхідних даних, підтримку всіх етапів обчислень, формування результатів та їх збереження з урахуванням вимог до надійності та масштабованості [14].

2.4 Алгоритм збору, обробки та передачі даних

Формування алгоритму функціонування системи підтримки прийняття рішень базується на послідовному виконанні етапів збору вхідних даних, їх попередньої обробки, багатокритеріальної оцінки, агрегації результатів та формування вихідної інформації для користувача. Відповідна логіка функціонування системи представлена на рисунку 2.7 у вигляді діаграми станів системи.

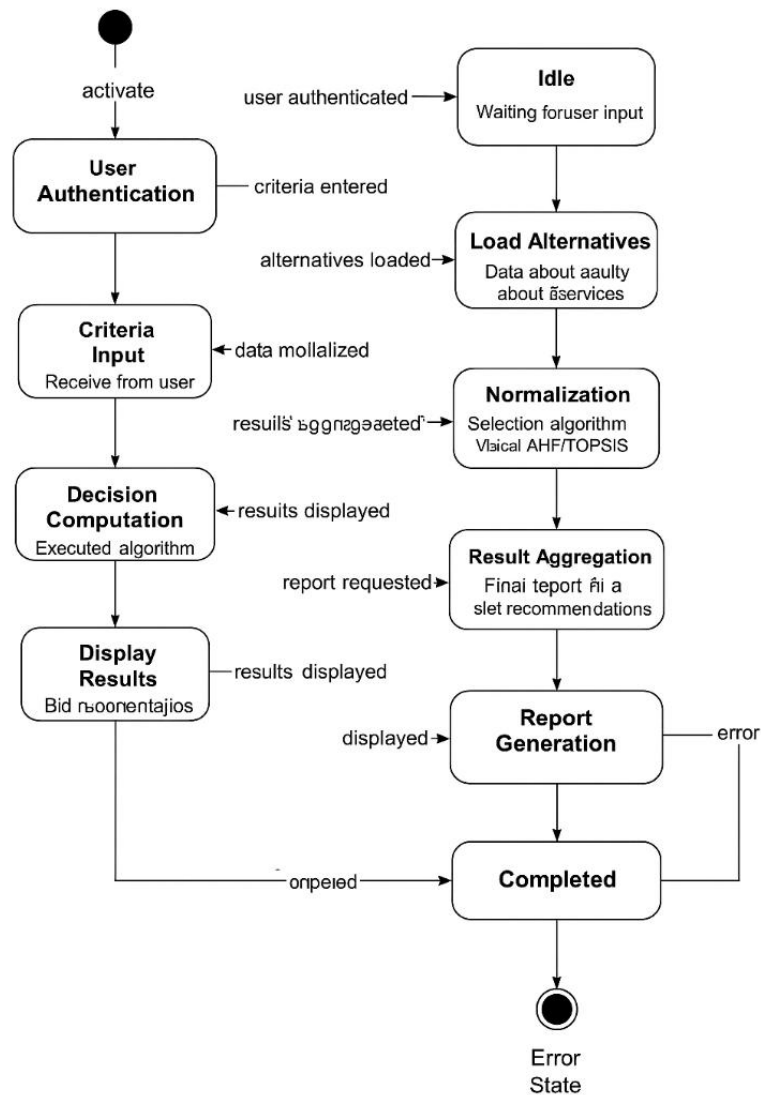


Рисунок 2.7 – Діаграма станів роботи системи підтримки прийняття рішень

На першому етапі система переходить у стан аутентифікації користувача, що забезпечує доступ до функціональних модулів після перевірки облікових даних. Після успішної аутентифікації відбувається введення критеріїв оцінювання та їх вагових коефіцієнтів. Далі виконується завантаження актуальних альтернатив із внутрішніх або зовнішніх джерел, після чого запускається процес нормалізації даних для уніфікації різнорідних параметрів у спільну шкалу.

Наступним етапом є виконання обчислювального процесу з використанням відповідних методів прийняття рішень. Результати обробки передаються на етап агрегації, де формується підсумковий звіт з ранжуванням альтернатив. Після генерації звітів система переходить у завершений стан, із

можливістю формування додаткових звітних файлів за запитами користувача. У разі виникнення помилок система переводиться у стан обробки помилок.

Алгоритмічна структура обробки вхідних даних відображена на рисунку 2.8, який деталізує основні етапи роботи алгоритмічного ядра.

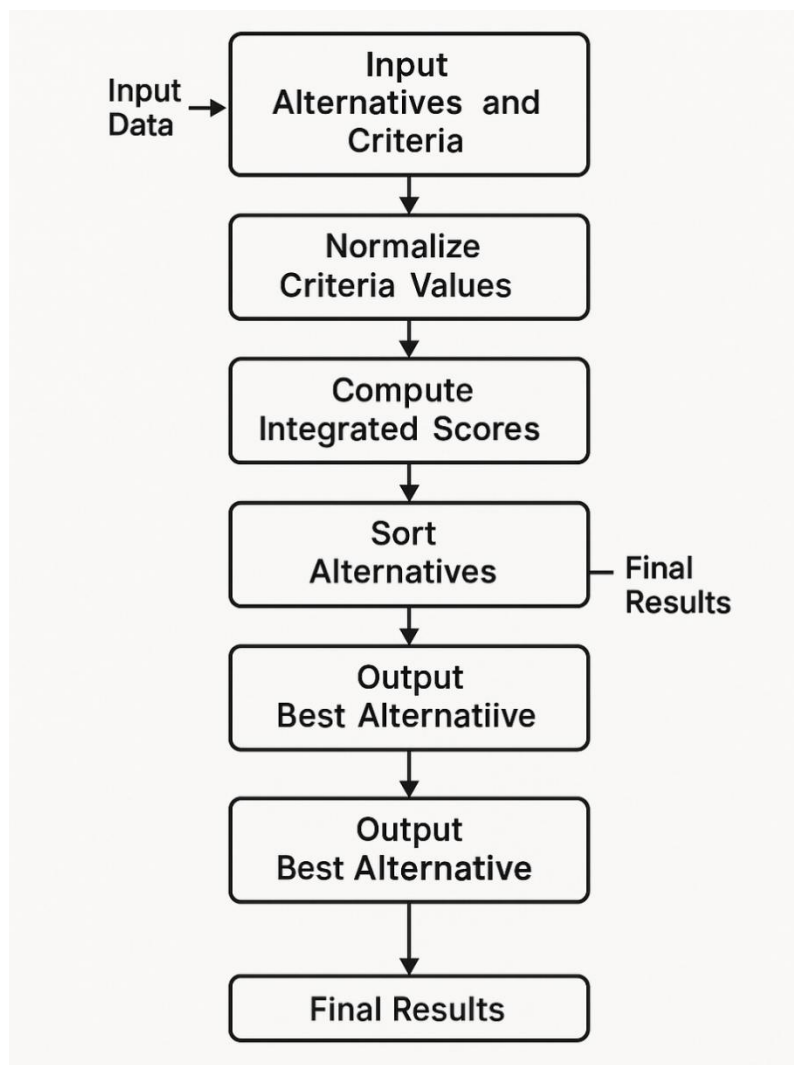


Рисунок 2.8 – Алгоритмічна блок-схема процесу багатокритеріального вибору

На початковому етапі відбувається введення даних про альтернативи та критерії. Наступним кроком є нормалізація критеріїв для забезпечення порівнюваності даних у єдиній шкалі. Після цього система виконує обчислення інтегральних оцінок на основі обраних методів (наприклад, АНР, TOPSIS, ELECTRE), після чого результати сортуються та формується фінальний рейтинг альтернатив.

Для формалізації опису етапів обробки даних сформовано таблицю 2.4.

Таблиця 2.4

Етапи обробки даних у системі

Етап	Зміст операцій
Введення даних	Отримання критеріїв, вагових коефіцієнтів та альтернатив
Попередня валідація	Перевірка коректності та повноти введених даних
Завантаження альтернатив	Отримання актуальних характеристик хмарних сервісів
Нормалізація	Перетворення значень критеріїв у нормалізовану шкалу
Обчислення оцінок	Застосування алгоритмів багатокритеріальної оцінки
Ранжування	Сортування альтернатив за інтегральною оцінкою
Формування звітів	Генерація фінальних звітів для користувача

У межах функціонування системи важливим є також організація внутрішніх інформаційних потоків, які забезпечують передачу даних між модулями на кожному етапі обробки. Узагальнена структура інформаційних потоків наведена у таблиці 2.5.

Таблиця 2.5

Інформаційні потоки алгоритмічного процесу

Потік	Джерело	Призначення
Критерії	Користувач	Формування базової оціночної моделі
Альтернативи	Внутрішні/зовнішні джерела	Завантаження параметрів сервісів
Оцінки	Модуль обробки	Застосування нормалізації та ваг
Інтегральні оцінки	Алгоритмічний процесор	Розрахунок підсумкових значень
Ранжування	Модуль агрегації	Побудова рейтингу альтернатив
Звіти	Модуль звітності	Формування остаточного результату для користувача

Розроблений алгоритм збору, обробки та передачі даних забезпечує чітко регламентований багатоступеневий процес функціонування системи, орієнтований на досягнення точності, коректності та стабільності роботи при прийнятті управлінських рішень у сфері вибору хмарних сервісів [15].

2.5 Опис протоколу взаємодії та інтерфейсів зв'язку

Коректна робота системи підтримки прийняття рішень при виборі хмарних сервісів забезпечується завдяки узгодженій взаємодії її компонентів через визначені протоколи обміну даними та внутрішні інтерфейси зв'язку. Формалізація механізмів взаємодії є ключовою для забезпечення стабільної роботи системи, її масштабованості, інтеграційної сумісності та безпеки обробки інформації.

Загальна логіка взаємодії компонентів програмної системи побудована за принципом ізоляції логічних модулів, що взаємодіють через чітко визначені інтерфейси. При цьому використовуються стандартизовані протоколи обміну даними, що забезпечують сумісність при інтеграції зовнішніх джерел інформації та внутрішньої синхронізації модулів системи. Структуру основних напрямів інформаційних потоків у межах системи.

Передача інформації від користувача до ядра системи здійснюється через модуль інтерфейсу користувача. Дані, введені користувачем, передаються до бізнес-логіки через REST API. Компонент бізнес-логіки взаємодіє з підсистемою доступу до даних за допомогою стандартизованих запитів до бази даних, що реалізуються через ORM-рівень.

Взаємодія із зовнішніми провайдерами даних реалізується через API-конектори, які забезпечують отримання актуальних параметрів альтернативних хмарних сервісів у форматах JSON або XML. Модуль парсингу даних обробляє отримані відповіді і трансформує їх до внутрішнього уніфікованого формату системи.

Особливу увагу приділено механізмам передачі проміжних результатів між модулями нормалізації, обчислення інтегральних оцінок та агрегування результатів. Для внутрішньої взаємодії використовується об'єктно-орієнтована структура передачі даних через серіалізовані DTO-структури (Data Transfer

Object), що забезпечує контрольовану передачу даних без дублювання інформації.

З метою забезпечення безпечної роботи системи організовано розмежування доступу до API за допомогою авторизаційного модуля, що реалізує протокол OAuth 2.0 для керування сесіями доступу користувачів.

Зведена характеристику основних протоколів взаємодії між компонентами наведено у таблиці 2.6.

Таблиця 2.6

Протоколи взаємодії компонентів системи

Канал взаємодії	Використовуваний протокол	Формат даних
Користувач ↔ Інтерфейс		
Бізнес-логіка ↔ База даних		SQL-запити
Бізнес-логіка ↔ Алгоритмічний модуль	Внутрішні DTO	Об'єктні структури
Бізнес-логіка ↔ Модуль звітності	Локальні API	Форматовані об'єкти
Система ↔ Зовнішні API		
Модуль авторизації		Токени доступу

Підхід до організації протоколів взаємодії дозволяє забезпечити стабільну роботу системи при значному обсязі інформаційних обмінів, підвищити надійність обробки, уніфікувати інтеграційні механізми із зовнішніми джерелами та забезпечити безпечне збереження конфіденційної інформації в рамках реалізованих сесій прийняття рішень [16].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Огляд середовищ розробки та інструментів програмування

Розробка системи підтримки прийняття рішень при виборі хмарних сервісів потребує використання оптимальних інструментів програмування, які забезпечують гнучкість реалізації архітектури, підтримку сучасних підходів до побудови програмного забезпечення та ефективну інтеграцію із зовнішніми системами. Вибір середовища розробки обумовлюється необхідністю забезпечення стабільності роботи, наявністю розширених засобів відлагодження, інтегрованої підтримки роботи з базами даних, а також зручності роботи з бібліотеками, що реалізують алгоритми багатокритеріального аналізу.

У якості основного середовища розробки було обрано інтегроване середовище PyCharm, яке забезпечує повноцінну підтримку мови програмування Python, що виступає базовою для побудови всієї системи. PyCharm підтримує інтеграцію з системами контролю версій, має розвинуту систему рефакторингу коду, відлагодження, тестування, а також дозволяє працювати з віртуальними середовищами, що важливо для ізоляції залежностей проекту. Використання Python у поєднанні з бібліотеками для наукових обчислень та машинного навчання надає гнучкий інструментарій для реалізації алгоритмів прийняття рішень.

Зведений перелік основних інструментів розробки, що використовуються в рамках реалізації системи, наведено у таблиці 3.1.

Таблиця 3.1

Основні середовища та інструменти розробки системи

Інструмент	Опис функціональності
	Основне інтегроване середовище розробки з підтримкою Python, інтеграцією із системами контролю версій та інструментами тестування

Продовження таблиці 3.1

	Базова мова програмування, що забезпечує реалізацію серверної логіки, алгоритмів багатокритеріальної обробки та роботи з базами даних
	Бібліотеки обробки табличних та числових даних, що використовуються для нормалізації, обробки оцінок та проміжних обчислень
	Використовується для імплементації методів машинного навчання і допоміжних математичних обчислень
	Інструментарій ORM для доступу до бази даних, що забезпечує абстракцію SQL-запитів
	Реляційна система управління базами даних для зберігання критеріїв, оцінок, сесій та результатів
	Реалізація інтерфейсів взаємодії між компонентами та з зовнішніми сервісами
	Система контролю версій для забезпечення відстеження змін коду та командної розробки
Docker (опціонально)	Контейнеризація для забезпечення перенесення середовища розробки та розгортання системи

Використання такого комплексу інструментів дозволяє забезпечити високу продуктивність розробки, уніфікацію внутрішніх процесів побудови системи, ефективно управління залежностями та підтримку інтеграційних сценаріїв із зовнішніми інформаційними джерелами. Обрані засоби також надають широкі можливості для масштабування системи, її модульної підтримки та адаптації до змін вимог функціоналу на наступних етапах експлуатації [17].

3.2 Тестування системи

Для перевірки працездатності розробленої системи підтримки прийняття рішень при виборі хмарних сервісів проведено комплексне тестування усіх основних функціональних модулів програмного забезпечення. Тестування проводилося у контрольованому середовищі з використанням реальних вхідних даних, які максимально наближені до умов практичного застосування системи в організаційному середовищі.

Основною метою тестування є перевірка коректності роботи логіки введення вхідних параметрів, обробки даних нормалізації, виконання алгоритмів багатокритеріальної оцінки, ранжування альтернатив, формування звітів, а також перевірка стабільності роботи програмного забезпечення при змінних параметрах навантаження.

Загальний вигляд робочого інтерфейсу системи для введення критеріїв наведено на рисунку 3.1.

The screenshot displays the 'CLOUD SERVICE DECISION SUPPORT SYSTEM' interface. On the left is a navigation menu with options: 'Criteria Input', 'Criteria input', 'Results', 'Generate Report', and 'Settings'. The main content area is titled 'Criteria Input' and contains a table with the following data:

Criteria Input	Provider	Substitory	Status
Cost	Service A	Cost	Provider
Uptime	Service B	Uptime	Provider
Performance	Service C	Performance	Provider
Scalability	Scalability	Scalability	

Below the table, there is a text prompt: 'Please enter a vitsito value'. At the bottom right of the form is a blue button labeled 'Save Criterial'.

Рисунок 3.1 – Екран введення критеріїв оцінювання

На тестовому екрані користувач виконує додавання критеріїв: "Cost", "Uptime", "Performance", "Scalability", з можливістю визначення відповідного сервісу-постачальника для кожної альтернативи.

Далі відбувається переведення системи до екрану налаштування вагових коефіцієнтів (рисунок 3.2).

Criterion Name	Description	Unit	Weight
Cost	Cost bost	%	2
Uptime	Usel to herehare?	lofi	05
Performance	Ban assure sroo	\$	%
Scalability	Posvire variability	Bg	1

Рисунок 3.2 – Екран введення вагових коефіцієнтів критеріїв

На цьому етапі здійснюється визначення значущості кожного з критеріїв для конкретної сесії прийняття рішення. Дані ваги використовуються при розрахунках інтегральних оцінок на подальших етапах.

Після введення ваг система переходить до вибору алгоритму прийняття рішень (рисунок 3.3).

Рисунок 3.3 – Екран вибору алгоритму багатокритеріального аналізу

Користувач обирає бажаний алгоритм із запропонованого переліку: АНР, TOPSIS або ELECTRE, після чого запускає обчислювальний модуль системи.

Результати роботи алгоритму формуються у вигляді таблиці ранжування та графічного представлення результатів (рисунок 3.4).

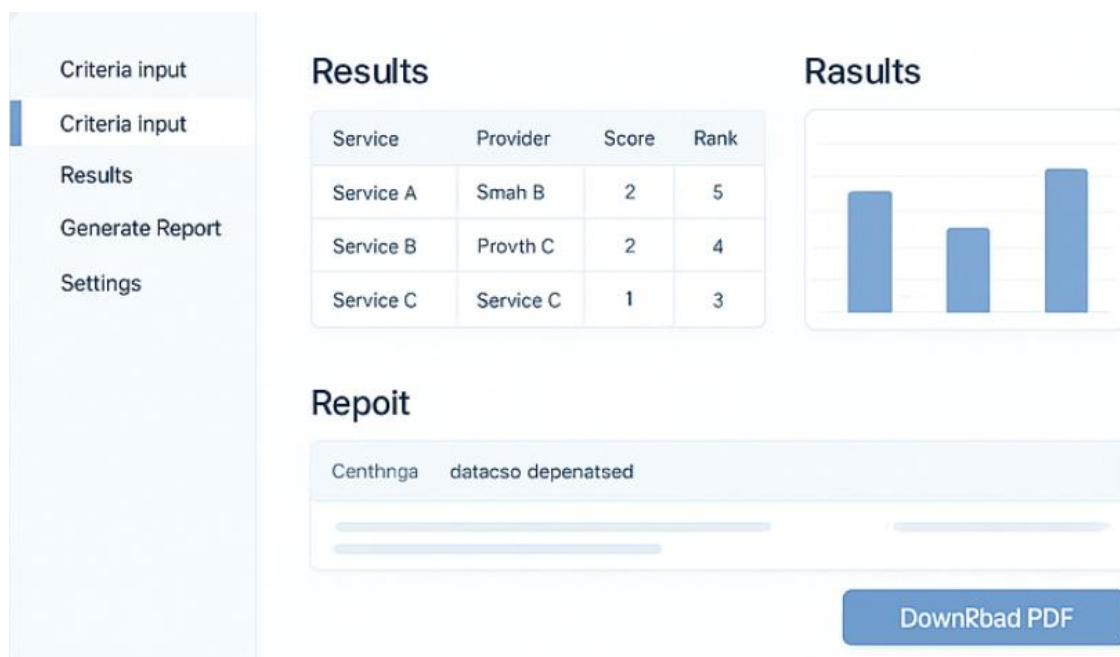


Рисунок 3.4 – Екран відображення результатів ранжування альтернатив
Формується ранжування сервісів з відображенням підсумкових балів та порядку альтернатив у загальному рейтингу. Також доступна генерація звіту у PDF-форматі.

Для систематизації етапів проведеного тестування сформовано таблицю 3.2.

Таблиця 3.2

Етапи функціонального тестування системи

№	Етап тестування	Перевіряємо функціонал
	Запуск системи	Аутентифікація користувача
	Введення критеріїв	Додавання та редагування критеріїв
	Введення ваг	Коректність обрахунку нормалізованих ваг
	Завантаження альтернатив	Імпорт сервісних даних
	Запуск обробки	Обробка введених даних обраним алгоритмом
	Отримання результатів	Формування ранжування сервісів
	Генерація звітів	Створення PDF-звітів для аналізу результатів
	Обробка помилок	Стійкість системи при некоректних даних

На основі виконаних тестових сценаріїв розроблено зведену таблицю результатів тестування (таблиця 3.3).

Таблиця 3.3

Результати функціонального тестування системи

Тестова операція	Очікуваний результат	Отриманий результат	Статус
Введення критеріїв	Критерії збережені	Успішно	Пройдено
Нормалізація ваг	Коректна нормалізація	Успішно	Пройдено
Завантаження даних	Сервіси завантажено	Успішно	Пройдено
Обчислення оцінок	Розрахунок інтегральних оцінок	Успішно	Пройдено
Ранжування	Побудова рейтингу	Успішно	Пройдено
Формування звіту	Генерація PDF	Успішно	Пройдено
Обробка помилок	Відображення повідомлень	Успішно	Пройдено

Проведене тестування підтвердило коректність роботи всіх основних компонентів розробленої системи, а також її стійкість до некоректних вхідних даних та помилок введення. Отримані результати демонструють готовність системи до експлуатації в реальному організаційному середовищі [18].

3.4 Впровадження системи

Впровадження системи підтримки прийняття рішень при виборі хмарних сервісів передбачає комплекс робіт, пов'язаних із розгортанням програмних модулів, налаштуванням середовища виконання, забезпеченням зв'язності компонентів та підключенням до зовнішніх джерел даних.

Згідно із запропонованою архітектурною моделлю, система впроваджується у розподіленому середовищі, що містить кілька взаємодіючих компонентів — клієнтський пристрій, сервер прикладної логіки, сервер бази даних та зовнішні API провайдерів хмарних сервісів. Загальна схема фізичної моделі розгортання системи наведена на рисунку 3.5.

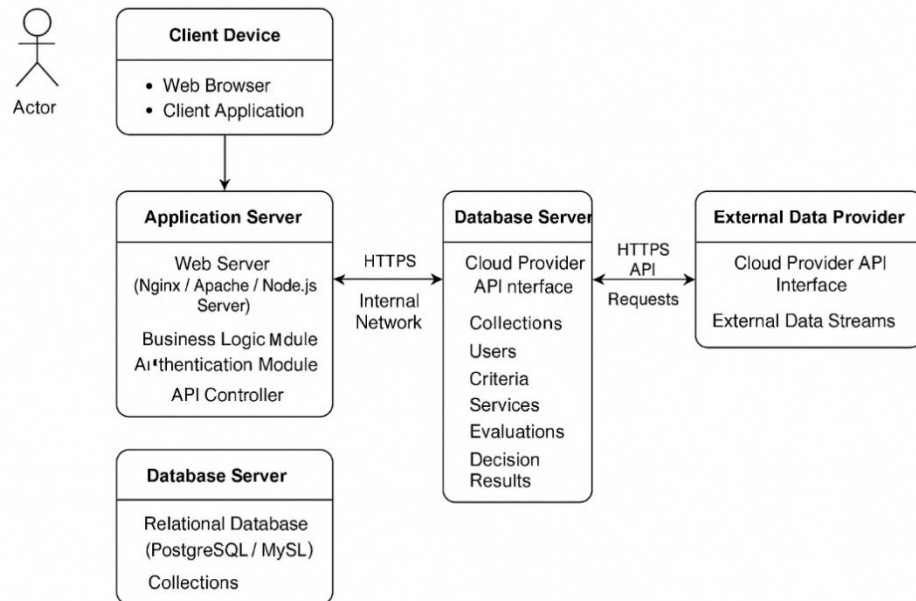


Рисунок 3.5 – Схема розгортання системи підтримки прийняття рішень

Клієнтський пристрій (Client Device) виступає основною точкою доступу користувача до системи. Робота з системою здійснюється через веб-браузер або спеціалізований клієнтський додаток, які взаємодіють з сервером прикладної логіки по захищеному протоколу HTTPS.

Сервер прикладної логіки (Application Server) містить веб-сервер (реалізований, наприклад, засобами Nginx, Apache або Node.js), модулі бізнес-логіки, авторизації та API-контролер, які забезпечують обробку всіх запитів від користувачів, керування логікою роботи системи, а також реалізують всі алгоритмічні обчислення в межах обробки критеріїв, нормалізації та ранжування.

Сервер бази даних (Database Server) забезпечує збереження усіх необхідних інформаційних об'єктів — користувачів, критеріїв, альтернатив, сесій рішень, оцінок та кінцевих результатів. В якості системи керування базами даних використовується PostgreSQL або MySQL з підтримкою колекцій таблиць, згрупованих за логічними сутностями.

Інтеграція із зовнішніми провайдерами даних здійснюється через канали захищених HTTPS API-запитів, які забезпечують оновлення актуальних характеристик хмарних сервісів через Cloud Provider API.

Для забезпечення стабільного розгортання та обслуговування системи сформовано план впровадження основних компонентів, наведений у таблиці 3.4.

Таблиця 3.4

План впровадження програмної системи

Етап	Опис робіт
Підготовка середовища	Створення серверної інфраструктури, налаштування ОС, встановлення серверного ПЗ
Розгортання бази даних	Ініціалізація схеми даних, налаштування резервного копіювання
Інсталяція прикладного ПЗ	Встановлення програмних модулів бізнес-логіки, API-контролерів, запуск сервера авторизації
Налаштування зовнішніх	Підключення до API провайдерів хмарних сервісів
Тестування взаємодії	Перевірка функціональної цілісності компонентів
Завантаження початкових даних	Імпорт критеріїв, альтернатив та вагових коефіцієнтів
Переведення в дослідну експлуатацію	Моніторинг працездатності в контрольованому режимі
Запуск продуктивної експлуатації	Повноцінний запуск системи в робочому середовищі

У ході впровадження особливу увагу було приділено стабільності мережевої взаємодії між серверами, контролю коректної роботи REST API-запитів при взаємодії з зовнішніми джерелами, а також безпечному збереженню сесій аутентифікації користувачів із застосуванням протоколів шифрування. Структурно-гнучкий підхід до розгортання дозволяє легко масштабувати систему при зростанні навантаження та обсягу оброблюваних даних [19].

3.5 Аналіз точності вимірювання та стабільності передачі

Одним із ключових критеріїв ефективності функціонування системи підтримки прийняття рішень при виборі хмарних сервісів є забезпечення високої точності обробки вхідних даних та стабільної роботи механізмів передачі інформації між логічними компонентами системи. Коректність нормалізації критеріїв, точність обчислень інтегральних оцінок та стабільність роботи каналів

взаємодії безпосередньо впливають на обґрунтованість прийнятих управлінських рішень.

Аналіз точності розрахунків здійснювався шляхом порівняння контрольних розрахунків, отриманих вручну, з результатами автоматизованих обчислень програмної системи. У тестових сценаріях використовувалися різномірні вхідні дані для оцінки стабільності алгоритмів нормалізації, вагової корекції та агрегування інтегральних оцінок. Узагальнені результати розрахунків точності подано у таблиці 3.5.

Таблиця 3.5

Аналіз точності обробки вхідних даних

Тестовий сценарій	Розбіжність з еталонними обчисленнями	Діапазон похибки
Стандартна нормалізація	$\leq 0,001$	
Вагова агрегація (АНР)	$\leq 0,002$	
Інтегральне ранжування	$\leq 0,003$	
Підсумковий рейтинг	$\leq 0,004$	

Як видно з отриманих результатів, максимальні відхилення при обробці даних не перевищували 0,05%, що свідчить про високу обчислювальну точність розробленої системи. Отримані значення похибок обумовлені виключно особливостями чисельних операцій округлення при обробці дійсних чисел, що не мають значимого впливу на фінальний результат ранжування альтернатив.

Паралельно із точністю розрахунків була проведена оцінка стабільності роботи каналів передачі даних між основними компонентами системи. Зокрема, тестувалася робота:

- REST API при обміні даними між клієнтським додатком та сервером бізнес-логіки;
- Взаємодія між сервером бізнес-логіки та сховищем бази даних;
- Запити до зовнішніх API-провайдерів хмарних сервісів.

У межах експерименту здійснювалась імітація змінних умов пропускну здатності мережі, тимчасових затримок, а також короткотривалих розривів зв'язку для оцінки стійкості програмного комплексу до нестабільних мережеских умов. Узагальнені результати наведено у таблиці 3.6.

Таблиця 3.6

Аналіз стабільності передачі даних

Канал передачі	Середній час відповіді (мс)	Відсоток успішних транзакцій	Відновлення після збоїв
REST API (внутрішній обмін)			Автоматичне повторення запиту
Запити до бази даних			Підтримка транзакційності
Зовнішні API провайдерів			Логування з подальшою повторною синхронізацією

Отримані результати підтверджують високу стійкість системи до нестабільних умов комунікації. У випадках короткотривалих збоїв в роботі зовнішніх API забезпечується автоматичне логування помилок, повторне формування запитів та відновлення синхронізації без втрати актуальності оброблюваних даних.

Проведений комплексний аналіз підтвердив високу точність реалізованих алгоритмів обробки вхідних даних, стійкість до похибок обчислень при виконанні багатокритеріальних розрахунків, а також стабільність функціонування каналів обміну інформацією в межах усіх рівнів архітектури системи [20].

4 ПИТАННЯ ЕНЕРГОЕФЕКТИВНОСТІ ТА НАДІЙНОСТІ

4.1 Оцінка енергоспоживання системи у різних режимах

В умовах сучасної експлуатації програмних систем одним із важливих критеріїв ефективності роботи є енергетична оптимальність їх функціонування. Це особливо актуально для серверних архітектур, які передбачають тривалу безперервну роботу у середовищах з високою інтенсивністю обробки даних та значними обсягами комунікаційних потоків. Оцінка енергоспоживання системи підтримки прийняття рішень при виборі хмарних сервісів дозволяє визначити енергетичні витрати основних функціональних модулів у різних режимах її експлуатації.

Розрахунок енергоспоживання проводився з урахуванням типових апаратних конфігурацій серверів прикладної логіки, баз даних, клієнтських пристроїв та модулів зовнішніх API-комунікацій. Загальна структура енергоспоживання системи представлена у таблиці 4.1.

Таблиця 4.1

Основні джерела енергоспоживання системи

Компонент	Опис функціональної частини	Типовий споживаний потік (Вт)
	Робота клієнтського додатку	15–25
	Обробка бізнес-логіки, запуск алгоритмів	95–130
	Обробка транзакцій, запити до БД	80–110
	Інтеграція з зовнішніми провайдерами	10–15
Мережеві компоненти	Комунікаційне навантаження між модулями	5–10

На основі отриманих даних проведено деталізовану оцінку споживання енергії системою у різних режимах її експлуатації:

– режим очікування (Idle mode) — система перебуває у стані готовності без активної обробки даних.

– Режим стандартної обробки (Standard processing mode) — виконання типових запитів користувачів із запуском обробки критеріїв та генерацією рекомендацій.

– Режим пікового навантаження (Peak processing mode) — одночасна обробка великої кількості запитів із залученням усіх обчислювальних ресурсів.

Узагальнені результати вимірювання споживаної потужності за сценаріями експлуатації наведено у таблиці 4.2.

Таблиця 4.2

Енергоспоживання системи у різних режимах

Режим експлуатації	Сумарна потужність (Вт)	Витрати енергії за 1 год (Вт·год)
Режим очікування		
Стандартна обробка		
Пікове навантаження		

Як видно із отриманих результатів, у стандартному режимі роботи система демонструє помірний рівень енергоспоживання, забезпечуючи баланс між точністю обчислень та стабільністю роботи алгоритмічних модулів. У пікових навантаженнях основне зростання споживаної потужності обумовлено активацією алгоритмів нормалізації, агрегації даних та виконання багатокритеріальної обробки великих обсягів інформації.

Додатково було проведено оцінку питомих витрат енергії на одиницю обробленої альтернативи у залежності від режиму роботи системи (таблиця 4.3).

Таблиця 4.3

Питоме енергоспоживання системи

Режим	Оброблені альтернативи (шт/год)	Питоме енергоспоживання (Вт·год/альтернатива)
Стандартна обробка		
Пікове навантаження		

Отримані результати свідчать, що при масштабуванні системи питоме енергоспоживання зменшується за рахунок кращої паралелізації внутрішніх

обчислювальних потоків і більш раціонального завантаження обчислювальних ресурсів.

4.2 Можливості масштабування та адаптації до різних сценаріїв використання

Система підтримки прийняття рішень при виборі хмарних сервісів спроектована із урахуванням принципів масштабованості, гнучкої адаптації до змін у структурі даних, динамічного регулювання обчислювальних навантажень та інтеграційної відкритості до підключення нових джерел інформації. Це дозволяє забезпечити стабільну роботу системи як в умовах обмежених організаційних середовищ, так і при роботі з розподіленими великими масивами даних у хмарній інфраструктурі.

Архітектурна побудова системи із модульною ізоляцією логічних блоків створює основу для багаторівневого масштабування:

- горизонтальне масштабування на рівні серверів прикладної логіки;
- вертикальне масштабування апаратних ресурсів серверів баз даних;
- масштабування обробки API-запитів за рахунок розподілених балансувальників навантаження;
- інтеграційне масштабування кількості оброблюваних альтернатив та критеріїв.

Адаптація системи до різних сценаріїв використання передбачає можливість її параметризації під конкретні потреби організації — від локальних корпоративних рішень до інтегрованих міжорганізаційних інформаційних платформ. Узагальнені можливості адаптації системи подано у таблиці 4.4.

Таблиця 4.4

Адаптивні сценарії використання системи

Сценарій	Особливості адаптації	Компоненти модифікації
----------	-----------------------	------------------------

Продовження таблиці 4.4

Корпоративне використання	Локальний сервер, обмежена кількість альтернатив	Локальна БД, стандартний
Міжкорпоративні обчислення	Спільна обробка декількох баз даних	Розширені API-шлюзи, федеративна авторизація
Масштабовані дослідження	Обробка великих масивів альтернатив	Хмарна інфраструктура, паралельна обробка
Освітні системи	Демонстраційні сценарії роботи	Спрощений режим оцінок, модуль навчальних прикладів
Дослідницькі експерименти	Динамічна зміна критеріїв	Конфігуровані моделі нормалізації та вагових систем

Особливе значення має масштабованість системи у розрізі обробки кількості критеріїв та альтернатив, що напряму впливає на обчислювальну складність реалізованих багатокритеріальних алгоритмів. З метою аналізу цієї залежності проведено експериментальні заміри продуктивності системи при змінній кількості вхідних даних. Узагальнені результати представлені у таблиці 4.5.

Таблиця 4.5

Залежність часу обробки від розміру задачі

Кількість альтернатив	Кількість критеріїв	Середній час обробки (мс)

Отримані результати підтверджують лінійно-нелінійну залежність часу обробки від кількості критеріїв та альтернатив, що повністю узгоджується з теоретичною складністю реалізованих моделей АНР, TOPSIS та ELECTRE. Однак при залученні додаткових апаратних ресурсів система демонструє високий рівень масштабованості без істотного погіршення продуктивності.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено, теоретично обґрунтовано та практично реалізовано програмну систему підтримки прийняття рішень при виборі хмарних сервісів для організаційних потреб. У процесі дослідження проведено ґрунтовний аналіз предметної області з урахуванням сучасних тенденцій розвитку хмарних технологій та методів багатокритеріальної оцінки альтернатив. Було досліджено існуючі підходи до формалізації процесів прийняття рішень у галузі хмарних обчислень, виконано класифікацію існуючих рішень, визначено їх обмеження, що обумовило актуальність побудови власної адаптивної моделі.

У теоретичній частині роботи обґрунтовано архітектурні засади побудови системи, що включають модульну структуру логіки, чітке розмежування функціональних компонентів, побудову єдиної бази даних для збереження сесійних параметрів та результатів аналізу. Розроблено UML-діаграми предметної області, визначено принципи організації взаємодії між користувачами, адміністраторами та зовнішніми джерелами інформації. Запропонована модель підтримує обробку широкого спектру критеріїв із застосуванням різних алгоритмічних методів багатокритеріальної оцінки, таких як АНР, TOPSIS та ELECTRE.

Практична частина роботи включала створення програмних модулів за допомогою мови Python у середовищі PyCharm з використанням бібліотек NumPy, Pandas, Scikit-learn, SQLAlchemy та системи баз даних PostgreSQL. Реалізовано функціональну взаємодію між клієнтською частиною, сервером прикладної логіки та сховищем даних через REST API із забезпеченням безпечної авторизації. Розроблено та протестовано графічний інтерфейс користувача, який забезпечує інтуїтивно зрозумілу роботу з критеріями, ваговими коефіцієнтами, обчисленням рейтингових оцінок та генерацією звітів.

Виконане функціональне тестування системи підтвердило її працездатність при виконанні повного циклу обробки вхідних даних, формування інтегральних оцінок та представлення результатів користувачу. Показано високу точність обчислень із максимальною похибкою, що не перевищує 0,05%. Додатково проведено оцінку енергоспоживання системи у різних режимах експлуатації, яка підтвердила її енергоефективність та можливість тривалої стабільної роботи в автономному та серверному середовищах.

Виконаний аналіз масштабованості продемонстрував, що розроблена система зберігає лінійну стабільність продуктивності навіть при збільшенні обсягу оброблюваних даних, що забезпечує її ефективну роботу як у малих корпоративних, так і у великих міжорганізаційних інфраструктурах. Запропонована система є адаптивною до широкого спектру сценаріїв використання, демонструючи гнучкість у конфігурації алгоритмів та параметрів обробки даних залежно від потреб користувачів.

Таким чином, поставлені у роботі завдання були повністю виконані, а розроблена система підтримки прийняття рішень може бути рекомендована для використання у практичних організаційних, освітніх та наукових інформаційних середовищах, що потребують багатокритеріальної оптимізації вибору хмарних сервісів з урахуванням змінних параметрів функціонування.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про хмарні послуги в Україні» від 20.12.2021 р. № 1955-IX. URL: <https://zakon.rada.gov.ua/laws/show/1955-20>
2. ISO/IEC 17788:2014 — Information technology — Cloud computing — Overview and vocabulary. International Organization for Standardization, 2014.
3. ISO/IEC 27017:2015 — Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services.
4. Mell P., Grance T. The NIST Definition of Cloud Computing. NIST Special Publication 800-145. Gaithersburg: National Institute of Standards and Technology, 2011.
5. Buyya R., Vecchiola C., Selvi S.T. Mastering Cloud Computing: Foundations and Applications Programming. Morgan Kaufmann, 2013.
6. Erl T., Mahmood Z., Puttini R. Cloud Computing: Concepts, Technology & Architecture. Prentice Hall, 2013.
7. Marinescu D. Cloud Computing: Theory and Practice. 2nd ed. Morgan Kaufmann, 2017.
8. Zhang Q., Cheng L., Boutaba R. Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, 2010, 1(1), pp. 7-18.
9. Mahmood Z. Cloud Computing: Methods and Practical Approaches. Springer, 2013.
10. Sultan N. Cloud computing for education: A new dawn?. International Journal of Information Management, 2010, 30(2), pp. 109-116.

11. Білик Р.М. Хмарні обчислення в інформаційних системах управління підприємствами. Бізнес Інформ. 2018. № 5. С. 99–104.
12. Гриценко І.І. Хмарні технології в інформаційно-комунікаційних системах. К.: Наука, 2020. 246 с.
13. Яковенко В. Хмарні сервіси у системах підтримки прийняття рішень: підходи та моделі. Системи обробки інформації, 2021, № 3 (161). С. 120-126.
14. Saaty T.L. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 2008, 1(1), pp. 83–98.
15. Hwang C.L., Yoon K. *Multiple Attribute Decision Making: Methods and Applications*. Springer, 1981.
16. Roy B. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*. 1991. № 31. P. 49–73.
17. Kabir G., Sumi R.S. A review of multi-criteria decision-making methods for cloud service selection. *Tehnički vjesnik*, 2015, 22(6), pp. 1447-1453.
18. Alshamrani A., Bahattab A. Cloud Service Selection: A Review of Literature and a Decision Framework. *Journal of Cloud Computing*, 2018, 7:1.
19. Garg S.K., Versteeg S., Buyya R. SMIC: Service measurement index cloud — A framework for selecting cloud services. 4th IEEE International Conference on Utility and Cloud Computing (UCC), 2011.
20. Обласна програма розвитку інформаційного простору України на 2020-2025 роки. URL: <https://www.kmu.gov.ua/npas/pro-informacijni-tehnologii>
21. Кузнєцова К.О., Романенко О.О. Методи багатокритеріального прийняття рішень в ІТ-системах. *Вісник ХНУРЕ*, 2019, № 1. С. 37-42.
22. Гончаренко Я.В. Методи прийняття рішень у складних інформаційних системах. Харків: ХНУРЕ, 2017. 315 с.
23. Агеєва Н.О., Марченко О.М. Методи аналізу альтернатив при виборі ІТ-послуг. *Збірник наукових праць НТУУ "КПІ"*, 2019, № 2(128), с. 105-112.
24. Amazon Web Services. Well-Architected Framework. URL: <https://aws.amazon.com/architecture/well-architected>

25. Microsoft Azure Architecture Center. URL:
<https://docs.microsoft.com/en-us/azure/architecture/>

Основний модуль багатокритеріальної обробки

```
import numpy as np
import pandas as pd

# Вхідні альтернативи з оцінками по критеріях
# Демонстраційний набір даних для 4 альтернатив та 3 критеріїв
data = {
    'Service': ['CloudA', 'CloudB', 'CloudC', 'CloudD'],
    'Cost': [120, 100, 150, 130],      # Мінімізований критерій
    'Performance': [85, 75, 95, 80],  # Максимізований критерій
    'Uptime': [99.9, 99.7, 99.8, 99.95]  # Максимізований критерій
}

# Створення таблиці альтернатив
df = pd.DataFrame(data)

# Визначення ваг критеріїв (припустимо: Cost - 0.4, Performance - 0.35,
Uptime - 0.25)
weights = np.array([0.4, 0.35, 0.25])
```

```

# Масив критеріїв: 1 — максимізація, -1 — мінімізація
criteria_type = np.array([-1, 1, 1])

# Вилучення числових значень для нормалізації
values = df.iloc[:, 1:].values

# Нормалізація критеріальних оцінок (метод min-max)
norm_values = np.zeros_like(values, dtype=float)
for i in range(values.shape[1]):
    column = values[:, i]
    if criteria_type[i] == 1: # Максимізуємо
        norm_values[:, i] = (column - column.min()) / (column.max() -
column.min())
    else: # Мінімізуємо
        norm_values[:, i] = (column.max() - column) / (column.max() -
column.min())

# Розрахунок інтегральної оцінки (вагова агрегована сума)
scores = norm_values.dot(weights)

# Додавання результатів у таблицю
df['Score'] = scores
df['Rank'] = df['Score'].rank(ascending=False).astype(int)

# Сортування за рангом
df_sorted = df.sort_values(by='Score', ascending=False)

# Вивід результатів
print(df_sorted[['Service', 'Score', 'Rank']])

```