

**Міністерство освіти і науки України**  
**Луцький національний технічний університет**  
**Факультет комп'ютерних та інформаційних технологій**  
**Кафедра комп'ютерної інженерії та охоронних систем**

**КВАЛІФІКАЦІЙНА РОБОТА**  
**ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗБОРУ ТА**  
**АНАЛІЗУ ДАНИХ ДЛЯ ПІДТРИМКИ ДІЯЛЬНОСТІ**  
**ГРОМАДСЬКОЇ ОРГАНІЗАЦІЇ**

**DESIGNING AN INFORMATION SYSTEM FOR COLLECTING**  
**AND ANALYZING DATA TO SUPPORT THE ACTIVITIES OF A**  
**PUBLIC ORGANIZATION**

спеціальність 126 Інформаційні системи та технології  
(шифр і назва спеціальності)

освітня програма «Інформаційні системи та технології охорони і безпеки»  
(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи ІСТОз-41  
**МАТЮШЕНКО Святослав Вадимович**

\_\_\_\_\_  
(підпис)

Керівник:  
к.т.н., доцент  
**ТЕРЛЕЦЬКИЙ Тарас Володимирович**

\_\_\_\_\_  
(підпис)

Кваліфікаційну роботу  
допущено до захисту  
«\_\_» \_\_\_\_\_ 2026 р.  
Гарант освітньої програми:  
к.т.н., доцент  
**ТЕРЛЕЦЬКИЙ Тарас Володимирович**

\_\_\_\_\_  
(підпис)

Луцьк – 2026 року

# ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет: комп'ютерних та інформаційних технологій

Кафедра: комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 126 Інформаційні системи та технології

Освітня програма: «Інформаційні системи та технології охорони і безпеки»

ЗАТВЕРДЖУЮ

Завідувач кафедри КІБ

к.т.н., доцент Терлецький Т. В.

«\_\_» \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*МАТЮШЕНКУ Святославу Вадимовичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Проектування інформаційної системи збору та аналізу даних для підтримки діяльності громадської організації

Керівник роботи: к.т.н., доцент Терлецький Тарас Володимирович

затверджені наказом закладу вищої освіти від «16» грудня 2025 р. № 529/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи: «30» травня 2026 р.

3. Вихідні дані до роботи: Методичні вказівки до виконання кваліфікаційної роботи бакалавра. наукові та електронні джерела з питань проектування інформаційних систем, баз даних, SQL-аналітики та підтримки прийняття рішень. Вимоги до інформаційної системи громадської організації. Дані про бенефіціарів, потреби, вразливості та надану допомогу. Програмні засоби PostgreSQL, Go, HTML/CSS, Bootstrap і Docker.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): Анотація. Вступ. Розділ 1. Аналітичний огляд стану предметної області (цифровізація діяльності громадських організацій; інформаційно-аналітичні системи підтримки прийняття рішень; методи багатокритеріального аналізу у задачах пріоритизації; методи організації та аналітичної обробки даних; аналіз існуючих програмних рішень та наукових підходів; постановка завдань на КРБ). Розділ 2. Обґрунтування вибору засобів та методів реалізації (загальна логіка проектування системи; аналіз вимог до системи; архітектура інформаційно-аналітичної системи; проектування моделі даних; проектування аналітичного конвеєра; проектування моделі підтримки прийняття рішень; пояснюваність та простежуваність результатів; валідація проектних рішень). Розділ 3. Практична реалізація (технологічне середовище реалізації; реалізація бази даних та аналітичного рівня; інтерфейс користувача для введення даних та подання результатів; реалізація backend та взаємодії з інтерфейсом користувача; інтеграція, перевірка та запуск системи; експериментальна перевірка та оцінка результатів). Загальні висновки та рекомендації. Список використаних джерел. Додатки.

5. Перелік графічного (ілюстративного) матеріалу: Презентація на 15 слайдах

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Розділ 1 Аналітичний огляд стану предметної області	<i>Терлецький Т. В.</i>		
Розділ 2 Обґрунтування вибору засобів та методів реалізації	<i>Терлецький Т. В.</i>		
Розділ 3 Практична реалізація	<i>Терлецький Т. В.</i>		
Загальні висновки та рекомендації	<i>Терлецький Т. В.</i>		
Нормоконтроль	<i>Кайдик О. Л.</i>		
Гарант ОП	<i>Терлецький Т. В.</i>		
Показник запозичень тексту			
Академічна доброчесність	<i>Кайдик О. Л.</i>		

7. Дата видачі завдання: «16» грудня 2025 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1.	Обґрунтування теми	До 12.12.2025 р.	
2.	Огляд літератури із досліджуваної проблеми	До 12.12.2025 р.	
3.	Розділ 1 Аналітичний огляд стану предметної області	До 28.02.2026 р.	
4.	Розділ 2 Обґрунтування вибору засобів та методів реалізації	До 31.03.2026 р.	
5.	Розділ 3 Практична реалізація	До 30.04.2026 р.	
6.	Загальні висновки та рекомендації	До 16.05.2026 р.	
7.	Формування списку використаних джерел	До 20.05.2026 р.	
8.	Формування додатків.	До 20.05.2026 р.	
9.	Формування презентації за темою кваліфікаційної роботи	До 20.05.2026 р.	
10.	Нормоконтроль	До 21.05.2026 р.	
11.	Інструментальна перевірка на академічний плагіат	До 22.05.2026 р.	
12.	Представлення кваліфікаційної роботи бакалавра до захисту	До 02.06.2026 р.	

Здобувач вищої освіти \_\_\_\_\_ (Матюшенко С. В.)  
(підпис)Керівник кваліфікаційної роботи \_\_\_\_\_ (Терлецький Т. В.)  
(підпис)

## АНОТАЦІЯ

Матюшенко С. В. Проектування інформаційної системи збору та аналізу даних для підтримки діяльності громадської організації. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інформаційні системи та технології охорони і безпеки». Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, загальних висновків та рекомендацій, списку використаних джерел та додатків.

Перший розділ присвячено аналізу предметної області діяльності громадських організацій. У ньому обґрунтовано актуальність теми, розглянуто проблеми управління даними, прозорості розподілу допомоги, інформаційно-аналітичні системи підтримки прийняття рішень, методи багатокритеріального аналізу та підходи до пріоритизації бенефіціарів.

У другому розділі здійснено вибір та обґрунтування методів і засобів реалізації проектних рішень. Визначено функціональні й нефункціональні вимоги, спроектовано архітектуру інформаційно-аналітичної системи, реляційну модель даних, аналітичний конвеєр, модель багатокритеріального оцінювання, а також механізми пояснюваності та валідації результатів.

Третій розділ присвячено практичній реалізації проектних рішень: реалізовано базу даних PostgreSQL, SQL-представлення для формування аналітичних ознак і оцінок пріоритетності, backend-компоненти, користувацький інтерфейс, засоби запуску та перевірки системи. Проведено експериментальну оцінку коректності результатів, стабільності ранжування, чутливості до зміни ваг критеріїв і відтворюваності обчислень.

Ключові слова: інформаційна система, громадська організація, база даних, PostgreSQL, SQL-аналітика, підтримка прийняття рішень, багатокритеріальний аналіз, пріоритизація, бенефіціар.

## ANNOTATION

Matiushenko S. Designing an Information System for Collecting and Analyzing Data to Support the Activities of a Public Organization. Manuscript.

Bachelor's qualification work EP «Security and safety information system and technologies». Lutsk National Technical University. Lutsk, 2026.

The qualification thesis consists of an introduction, three chapters, general conclusions and recommendations, a list of references, and an appendix.

The first chapter is devoted to the analysis of the subject area of non-governmental organizations. It substantiates the relevance of the topic and considers data management problems, transparency of aid distribution, information-analytical decision support systems, methods of multi-criteria analysis, and approaches to beneficiary prioritization.

The second chapter presents the selection and justification of methods and tools for implementing the design solutions. Functional and non-functional requirements are defined, and the architecture of the information-analytical system, relational data model, analytical pipeline, multi-criteria evaluation model, as well as mechanisms for explainability and validation of results are designed.

The third chapter is devoted to the practical implementation of the design solutions: the PostgreSQL database, SQL views for generating analytical features and priority scores, backend components, user interface, system launch tools, and verification mechanisms are implemented. The correctness of results, ranking stability, sensitivity to changes in criterion weights, and reproducibility of calculations are experimentally evaluated.

Keywords: information system, non-governmental organization, database, PostgreSQL, SQL analytics, decision support, multi-criteria analysis, prioritization, beneficiary.

## ЗМІСТ

ВСТУП .....	7
<b>РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ</b>	
1.1 Цифровізація діяльності громадських організацій .....	8
1.2 Інформаційно-аналітичні системи підтримки прийняття рішень .....	10
1.3 Методи багатокритеріального аналізу у задачах пріоритизації .....	12
1.4 Методи організації та аналітичної обробки даних .....	16
1.5 Аналіз існуючих програмних рішень та наукових підходів .....	18
1.6 Постановка завдань на кваліфікаційну роботу бакалавра .....	22
<b>РОЗДІЛ 2 ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ ТА МЕТОДІВ РЕАЛІЗАЦІЇ</b>	
2.1 Загальна логіка проектування системи .....	23
2.2 Аналіз вимог до системи .....	24
2.3 Архітектура інформаційно-аналітичної системи .....	25
2.4 Проектування моделі даних .....	27
2.5 Проектування аналітичного конвеєра .....	29
2.6 Проектування моделі підтримки прийняття рішень .....	31
2.7 Пояснюваність та простежуваність результатів .....	33
2.8 Валідація проектних рішень .....	35
<b>РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ</b>	
3.1 Технологічне середовище реалізації .....	38
3.2 Реалізація бази даних та аналітичного рівня .....	39
3.3 Інтерфейс користувача для введення даних та подання результатів ...	42
3.4 Реалізація backend та взаємодії з інтерфейсом користувача .....	46
3.5 Інтеграція, перевірка та запуск системи .....	47
3.6 Експериментальна перевірка та оцінка результатів .....	49
ЗАГАЛЬНІ ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ .....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	59

## ВСТУП

Громадські організації (ГО) часто мають обмежені ресурси та мусять підлаштовуватися під постійні зміни, тому для них важливо справедливо розподіляти допомогу. Проте, інформація про людей, які її потребують (бенефіціарів), зазвичай зберігається в різних хаотичних таблицях або файлах, через що важко порівнювати доходи людей, склад їх родин чи історію попередніх виплат.

Сучасні програми, які використовуються в ГО, підходять лише для простої реєстрації та обліку, але вони не можуть аналізувати дані та виділяти тих, кому допомога потрібна першочергово.

У результаті прийняття рішення про те, кому допомогти, часто приймають «вручну» і суб'єктивно. Основна технічна проблема полягає у тому, що наявні дані про бенефіціарів не зведено до єдиного стандарту (не нормалізовано), тому їх неможливо автоматично порівняти між собою або ж чітко перевірити, чому було прийняте те чи інше рішення.

Метою роботи є проектування та реалізація інформаційної системи збору, зберігання й аналітичної обробки даних для підтримки діяльності громадської організації та обґрунтованого визначення пріоритетності бенефіціарів.

Об'єкт проектування – процеси збору, зберігання, обробки та аналізу даних у діяльності громадських організацій, що пов'язані з підтримкою прийняття рішень щодо розподілу допомоги.

Предмет проектування – методи, моделі та програмні засоби аналізу даних і пріоритизації бенефіціарів на основі багатокритеріального оцінювання в інформаційній системі підтримки діяльності громадської організації.

## РОЗДІЛ 1

### АНАЛІТИЧНИЙ ОГЛЯД СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1 Цифровізація діяльності громадських організацій

Громадські організації забезпечують підтримку вразливих категорій населення в умовах обмежених ресурсів, змінних потреб і необхідності прозорого розподілу допомоги. Ефективність такої діяльності залежить від якості інформаційного забезпечення, оскільки дані про бенефіціарів, склад домогосподарства, доходи, потреби та історію допомоги є основою управлінських рішень.

«У контексті теорії управління соціальними системами інформація розглядається як чинник управління, вона необхідна на всіх етапах управлінської діяльності. Через інформацію здійснюється вплив на формування управлінських рішень, виявлення потреб у регулюванні конкретного виду суспільних відносин, їх реалізація, координація діяльності суб'єктів управління, контроль за реалізацією управлінських рішень тощо» [1].

Цифровізація соціальних послуг сприяє підвищенню доступності допомоги, скороченню часу реагування та зниженню організаційних витрат [2, с. 314], [3, с. 284]. Водночас для громадської організації важлива не лише автоматизація обліку, а й організація повного інформаційного циклу: збирання, зберігання, перевірка, аналіз та інтерпретація даних.

«Щоб здійснювати вплив, суб'єкт управління повинен мати певний обсяг відомостей, даних про об'єкт управління, знати особливості його можливої поведінки залежно від різної природи факторів тощо. Тобто суб'єкту управління потрібно мати інформацію про закономірності природи об'єкта управління (керованої системи). Але для того щоб мати таку інформацію, суб'єкту управління слід організувати її забезпечення, визначити обсяг, джерела її надходження, методи і способи збирання, аналізу, обробки, використання, зберігання тощо [1]».

Отже, актуальним є перехід від фрагментарного обліку до комплексної інформаційно-аналітичної системи, яка підтримує не тільки зберігання даних, а й прозоре оцінювання пріоритетності допомоги [4, с. 125], [5, с. 184].

Ключовою проблемою гуманітарних і соціальних програм є управління даними в умовах динамічного середовища, різномірності джерел і потреби у швидкому реагуванні [6]. На практиці дані часто формуються децентралізовано, що спричиняє дублювання, неповноту та логічні суперечності. Інформація із зовнішнього середовища, яка є часто приблизною, неточною, неповною, суперечливою, має ймовірнісний характер і через те вимагає нестандартних процедур опрацювання [6].

Для даних про бенефіціарів важливими є актуальність, повнота та логічна узгодженість. Недостатня інтеграція інформаційних ресурсів ускладнює координацію між організаціями [7, с. 38], а помилки в доходах, потребах або історії допомоги можуть викривляти пріоритизацію. Однією з причин низької ефективності цифрових трансформацій у соціальній сфері є відсутність адаптованих платформ для вразливих категорій населення [8, с. 657]. У цьому контексті цифрові технології здатні «суттєво покращити доступність соціальних послуг для різних груп населення» [9, с. 146].

Отже, прозорий розподіл допомоги потребує формалізованих критеріїв, перевірної логіки аналізу, контролю якості даних і детермінованих моделей оцінювання [1].

Окрему роль у цифровізації громадських організацій відіграють інформаційні системи, які у сфері соціальної підтримки переходять від простого обліку до аналітичної підтримки управлінських рішень. Для громадських організацій це означає потребу в системі, яка накопичує записи про бенефіціарів, формалізує оцінювання потреб, ранжує заявників і дає обґрунтування результатів.

Сучасні дослідження цифровізації соціальної сфери акцентують увагу на аналітичному використанні даних. «Концептуалізація проактивної моделі. Ця парадигма передбачає використання цифрових інструментів для багаторівневої

оцінки вразливості та розробки унікальних траєкторій відновлення» [10, с. 54]. У межах цієї роботи такий підхід реалізується через структурований збір даних, їх перевірку, SQL-орієнтовану обробку та формування оцінок пріоритетності.

## **1.2 Інформаційно-аналітичні системи підтримки прийняття рішень**

Інформаційно-аналітична система розглядається як інструмент підтримки аналізу, оцінювання та формалізованого прийняття рішень. Інтелектуальна інформаційна система трактується як система, здатна підтримувати процеси аналізу й прийняття рішень на основі формалізованих моделей обробки даних [11, с. 31]; у цій роботі інтелектуальність розуміється як структурована аналітична обробка, багатокритеріальне оцінювання та інтерпретованість результатів [12].

Архітектурно такі системи містять операційний рівень збору даних, рівень зберігання, аналітичний рівень і рівень представлення результатів користувачу [6, с. 46]. On-Line Analytical Processing (OLAP) – технологія оперативної аналітичної обробки даних, що використовує методи і засоби для збору, зберігання та аналізу багатовимірних даних з метою підтримки процесів прийняття рішень [12]. У цій роботі замість OLAP-кубів застосовано реляційні таблиці, SQL-представлення, агрегацію та формалізований розрахунок оцінок пріоритетності.

«Подібна архітектура дозволяє мінімізувати конфлікт між вимогами до швидкості транзакційної обробки та потребами аналізу. Іншою особливістю простору даних як інформаційно–аналітичної системи є необхідність попереднього налаштування для врахування особливостей предметної області, для якої він проектується. Такі особливості формуються у вигляді бази правил та змінюються у процесі експлуатації простору даних» [6].

Таким чином, у роботі інформаційно-аналітична система розуміється як середовище, що поєднує структуроване зберігання, SQL-орієнтовану обробку,

формалізоване оцінювання та подання результатів для підтримки управлінських рішень.

Окремим класом інформаційно-аналітичних систем є системи підтримки прийняття рішень СППР (Decision Support Systems, DSS), орієнтовані на аналітичну підтримку управління в умовах багатокритеріальності, неповноти інформації та необхідності оцінювання альтернатив [1]. У соціально-орієнтованих задачах вони важливі через відповідальність рішень щодо розподілу обмежених ресурсів.

Рівень потреби бенефіціара залежить від економічного стану, складу домогосподарства, соціальної вразливості, інтенсивності потреб та історії попередньої допомоги. Тому однокритеріальне оцінювання є недостатнім, а доцільним є багатокритеріальний аналіз із нормалізацією різнорідних показників і ваговим агрегуванням. Детальніше відповідні процедури розглянуто нижче.

Подібний підхід забезпечує прозорість оцінювання та можливість зміни політики прийняття рішень через ваги критеріїв. «Модифікація знань здійснюється за допомогою індуктивної компоненти (досвіду набутого системою у процесі свого функціонування)» [11, с. 48]. «Аналіз чутливості здійснює перевірку впливу змін вхідних даних або параметрів на передбачуване рішення, тобто результуючу змінну. Аналіз чутливості важливий, оскільки він дає можливість гнучкості й адаптації до змін умов і до вимог різних ситуацій ухвалення рішень» [12].

Оскільки показники мають різні масштаби вимірювання, їх безпосереднє агрегування є некоректним. Для приведення критеріїв до єдиного інтервалу застосовується нормалізація, тобто стандартизація й уніфікація [13]. «Наприклад, попереднє оброблення даних, побудова математичних та статистичних моделей, визначення їх адекватності, перетворення до стандартизованих форм (простір станів) ...» [12]. Отже, DSS у цій роботі розглядається як інструмент аналітичної підтримки, а не як заміна управлінського рішення.

Пояснюваність аналітичних моделей є ключовою вимогою до систем підтримки прийняття рішень у сфері соціальної підтримки. Вона означає здатність системи надати формалізоване обґрунтування результату оцінювання [14], а інтерпретованість - можливість користувача зрозуміти логіку впливу окремих факторів на кінцевий результат. Уніфікація аналізу припускає створення типових методик і інструкцій, вихідних форм і таблиць, стандартних програм, єдиних критеріїв оцінки, що забезпечує порівнянність, зведення результатів аналізу на більш високому рівні управління, підвищує об'єктивність оцінки діяльності підрозділів, зменшує витрати часу на аналіз і в підсумку сприяє підвищенню його ефективності [1, с. 85].

Для соціально-орієнтованих інформаційних систем непрозорі моделі прийняття рішень створюють ризики для аудиту, довіри та коригування політик розподілу допомоги. Тому в роботі використовується детермінована вагова модель, у якій логіка формування інтегральної оцінки може бути явно простежена [11]. Внесок критерію подається у вигляді (1.1)

$$C_i = w_i \cdot x_i, \quad (1.1)$$

де  $C_i$  – внесок  $i$ -го критерію;

$w_i$  – вага внеску  $i$ -го критерію;

$x_i$  – нормалізоване значення.

Сумарна оцінка формується як сума таких внесків, що дає змогу пояснити причини високого або низького рівня пріоритетності.

### **1.3 Методи багатокритеріального аналізу у задачах пріоритизації**

Задачі розподілу соціальної допомоги є слабкоструктурованими: вони містять кілька критеріїв, неповну інформацію та потребу врахування соціально-економічного контексту. Пріоритетність бенефіціара не може визначатися

одним показником, тому застосування методів багатокритеріального прийняття рішень (Multi-Criteria Decision Analysis, MCDA) є обґрунтованим підходом до формалізації пріоритизації [12].

Багатокритеріальний аналіз передбачає побудову моделі, у якій множина альтернатив  $A = \{a_1, a_2, \dots, a_n\}$  оцінюється за сукупністю критеріїв  $C = \{c_1, c_2, \dots, c_m\}$ . Кожна альтернатива характеризується вектором значень  $a_i = (x_{i1}, x_{i2}, \dots, x_{im})$ . У цій роботі альтернативами є бенефіціари або домогосподарства, а критеріями – показники потреби, вразливості, економічного стану та попередньо отриманої допомоги [15].

MCDA використовується для підтримки вибору в умовах конфлікту критеріїв і множинності допустимих рішень. Для соціальних програм це важливо, оскільки критерії мають різну інтерпретацію та неоднакову значущість [12]. У сучасних інформаційно-аналітичних системах MCDA дає змогу формалізувати політику прийняття рішень, забезпечити прозорість оцінювання, контроль впливу критеріїв і пояснення результатів ранжування.

Для практичного використання багатокритеріального оцінювання однією з ключових проблем є неоднорідність критеріїв. Без приведення показників до єдиного масштабу їх агрегування є некоректним [12]. Для задачі пріоритизації використовується min-max нормалізація, яка обмежує результат інтервалом  $[0; 1]$  і є сумісною з лінійними ваговими моделями.

Для критеріїв прямої дії використовується формула min-max нормалізації (1.2):

$$x_i^{\text{norm}} = (x_i - x_{\min}) / (x_{\max} - x_{\min}), \quad (1.2)$$

де:  $x_i^{\text{norm}}$  – нормалізоване значення і-го критерію;

$x_i$  – початкове значення критерію;

$x_{\min}$  та  $x_{\max}$  – мінімальне та максимальне значення критерію в наборі даних.

Формула (1.2) застосовується до показників, зростання яких збільшує пріоритетність.

Для критеріїв оберненої дії використовується інверсна форма нормалізації (1.3):

$$x_i^{\text{norm}} = 1 - [(x_i - x_{\min}) / (x_{\max} - x_{\min})]. \quad (1.3)$$

На відміну від формули (1.2), формула (1.3) зменшує нормалізоване значення зі зростанням початкового показника. Такий підхід використовується для доходу або обсягу раніше отриманої допомоги.

Методологічно важливо враховувати вироджені ознаки, коли  $x_{\max} = x_{\min}$ , оскільки знаменник формул (1.2) та (1.3) дорівнює нулю. У реалізованій системі це враховується через механізм NULLIF у SQL-поданнях. Відсутні значення також потребують окремої обробки, адже в гуманітарних системах неповнота даних є типовою ситуацією [6]; у роботі використано правило  $\text{NULL} \rightarrow 0$ , що забезпечує детермінованість обчислень.

Після нормалізації окремі критерії інтегруються в єдину функцію пріоритету. Одним із найбільш поширених підходів у системах підтримки прийняття рішень є лінійна адитивна модель (1.4) [9]:

$$S(a_i) = \sum w_j \cdot x_{ij}, \quad (1.4)$$

де:  $S(a_i)$  – інтегральний пріоритет альтернативи  $a_i$ ;

$w_j$  - вага  $j$ -го критерію;

$x_{ij}$  – нормалізоване значення  $j$ -го критерію для альтернативи  $a_i$ .

Для забезпечення інтерпретованості ваги нормуються за (1.5):

$$\sum w_j = 1. \quad (1.5)$$

Умова (1.5) дозволяє інтерпретувати ваги як частки впливу критеріїв на підсумкову оцінку, а внесок критерію подати як (1.6):

$$c_j = w_j \cdot X_j. \quad (1.6)$$

У представленій системі на базовому етапі використовується рівноваговий підхід, за якого п'ять критеріїв мають однакову вагу  $w_j=0,2$ . «Стратегія цифровізації та запровадження інформаційних технологій в системі соціального захисту орієнтується на задоволення потреб людини, визначених цільових категорій, а також на реалізацію сучасного інструментарію інформаційних систем для автоматизації рутинних процедур надання соціальних послуг, забезпечення соціального захисту, можливості виконання самостійних запитів на інформацію та виплати за допомогою веб-порталів» [16]. Обмеженням лінійної моделі є припущення про незалежність критеріїв, однак для прозорості системи підтримки рішень її інтерпретованість є суттєвою перевагою.

Після обчислення інтегральних оцінок за лінійною ваговою моделлю (1.4) виконується ранжування альтернатив (1.7):

$$a_{(1)} > a_{(2)} > \dots > a_{(n)}, \quad S(a_{(1)}) \geq S(a_{(2)}) \geq \dots \geq S(a_{(n)}). \quad (1.7)$$

У задачах гуманітарної підтримки таке ранжування використовується для визначення пріоритетності надання допомоги, формування списків очікування та розподілу обмежених ресурсів.

У реалізованій системі підсумковий результат додатково подається через категорії HIGH PRIORITY, MEDIUM PRIORITY та LOW PRIORITY. Категоризація спрощує інтерпретацію оцінок, однак завжди відображає певну політику розподілу ресурсів, тому важливими є зміна ваг критеріїв та аналіз наслідків такої зміни [17]. На рівні реалізації ранжування виконується засобами SQL через ORDER BY та віконну функцію RANK(). Аналіз чутливості полягає у

варіюванні ваг  $w_j \rightarrow w_j'$  і порівнянні оцінок, рангових позицій та кореляції між базовим і альтернативним ранжуванням [12].

#### 1.4 Методи організації та аналітичної обробки даних

Організація даних є основою інформаційної системи підтримки прийняття рішень, оскільки структура даних визначає можливості аналітичної обробки, узгодженість інформаційних потоків і достовірність результатів. Оскільки система працює з персональними та соціально чутливими даними, важливими є конфіденційність і контроль доступу відповідно до принципів GDPR [12].

Найбільш поширеною моделлю організації даних в інформаційно-аналітичних системах залишається реляційна модель Е. Кодда [13]. Вона подає інформацію у вигляді взаємопов'язаних таблиць, підтримує декларативну обробку даних і забезпечує контроль цілісності. Використання первинних і зовнішніх ключів зменшує ризики дублювання записів і втрати зв'язків між даними [13].

У реалізованій системі базовою одиницею аналізу є бенефіціар, представлений таблицею `beneficiaries` з атрибутами `id`, `household_size`, `monthly_income` та `dependents_count`. Додаткові характеристики зберігаються в таблицях `need_records`, `vulnerability_records` та `assistance_records`, що дозволяє підтримувати нормалізовану структуру й історичність даних через часові атрибути.

Важливою властивістю реляційних аналітичних систем є побудова похідних подань (`views`), які відокремлюють операційний рівень від аналітичного. «Віртуальна таблиця створюється для кожного з підтипів з тією метою, щоб надати можливість доступу лише до необхідних даних згідно з вимогами прикладних задач, що визначають спосіб маніпулювання сутностями, які є підтипами» [13].

У межах цієї роботи використовується багатошарова організація обробки: операційний рівень накопичує первинні дані, аналітичний рівень виконує агрегацію, нормалізацію та обчислення оцінок, а рівень представлення подає результати користувачу. Каскадні SQL-подання утворюють послідовність:

$$\text{raw} \rightarrow \text{normalized} \rightarrow \text{scores} \rightarrow \text{decisions},$$

у межах якої дані нормалізуються за формулами (1.2) і (1.3), перетворюються на інтегральні оцінки за формулою (1.4) та використовуються для формування рішень.

На основі реляційної моделі SQL у цій роботі використовується як механізм аналітичної обробки реляційних даних: агрегації показників, побудови похідних ознак, нормалізації критеріїв, обчислення інтегральних оцінок, ранжування альтернатив і статистичного контролю результатів. Аналітична обробка реалізується за допомогою агрегатних функцій, віконних функцій, вкладених запитів, спільних табличних виразів та SQL-подань. Наприклад, агрегація соціальних характеристик може виконуватися через `AVG()`, `SUM()` та `COUNT()`, а ранжування – через `RANK() OVER (ORDER BY score DESC)` [13].

Перевагою SQL-орієнтованого підходу є декларативність: логіка обчислень задається у вигляді запитів і подань, тому її можна перевірити, повторити й модифікувати. У системі аналітична логіка представлена як послідовність каскадних подань:

$$\text{VIEW}_1 \rightarrow \text{VIEW}_2 \rightarrow \text{VIEW}_3,$$

а статистичний контроль виконується через `AVG(score)`, `STDDEV(score)` та `corr(x_i, score)`, а пояснення пріоритету може формуватися конструкцією `CASE WHEN`.

У системах підтримки прийняття рішень візуалізація спрощує інтерпретацію результатів аналітичного оцінювання та підтримує управлінський аналіз [6]. У реалізованій системі це забезпечується веб-інтерфейсом на основі Go, HTML-шаблонів та Bootstrap [19]: результати

подаються у вигляді таблиць ранжування, гістограм розподілу, індикаторів кореляції, профілів бенефіціарів та аналізу зміни рангу.

Поряд із візуальним поданням результатів важливою є їх валідація, оскільки якість аналітичних результатів залежить від достовірності, повноти та узгодженості вхідних даних. У системах соціальної підтримки помилки в даних можуть викривляти пріоритети, призводити до необґрунтованого розподілу допомоги та порушувати принципи справедливості [12]. Навіть коректна аналітична модель не забезпечує достовірності результатів за умови використання неповних або суперечливих даних [6].

На рівні реляційної бази даних контроль забезпечується обмеженнями CHECK, первинними та зовнішніми ключами, а також обмеженнями NOT NULL. Наприклад, для рівнів потреб і вразливості можуть встановлюватися умови  $1 \leq severity \leq 5$  та  $1 \leq urgency \leq 5$ , що унеможлиблює запис значень поза визначеною шкалою оцінювання.

Аналітична валідація передбачає перевірку результатів нормалізації: формули (1.2) та (1.3) повинні формувати значення в межах інтервалу  $[0; 1]$ , тобто  $0 \leq x_i^{norm} \leq 1$ . Додатково контролюються вироджені ознаки, відсутні записи, порожні результати агрегування та аномальні розподіли. Статистичний контроль може виконуватися засобами SQL через  $MIN(score)$ ,  $MAX(score)$ ,  $STDDEV(score)$ ,  $corr(x_i, score)$  та аналіз  $distribution(score)$ .

Оскільки інтегральний бал формується за лінійною ваговою моделлю (1.4), кожний результат може бути незалежно перевірений і відтворений. Візуальне подання результатів додатково допомагає аналізувати розподіл оцінок, вплив критеріїв і пояснюваність прийнятих рішень [11].

## 1.5 Аналіз існуючих програмних рішень та наукових підходів

Інформаційні системи соціальної підтримки в діяльності неурядових організацій орієнтовані на облік бенефіціарів, координацію гуманітарної допомоги, моніторинг потреб населення та підтримку управлінських рішень. Їх

ефективність залежить від централізованого збору, структуризації та аналітичної інтерпретації даних [6]. У практиці державного й гуманітарного сектору поширені реєстрові системи, які накопичують соціально-економічну інформацію про домогосподарства, історію отримання допомоги та статус вразливості [16]. Однак значна частина таких рішень зосереджена на операційному обліку, а не на аналітичному оцінюванні [16].

Особливістю гуманітарних інформаційних систем є необхідність інтегрувати демографічні, фінансові, часові та соціальні дані [20]. Тому такі системи доцільно розглядати як поєднання транзакційного рівня, що накопичує операційні записи, та аналітичного рівня, що формує інтегровані показники для оцінювання потреб, ранжування бенефіціарів і сценарного аналізу розподілу ресурсів [12].

Для таких систем автоматизація оцінювання потреб бенефіціарів базується на формалізації критеріїв соціальної вразливості та їх агрегуванні в єдину оцінку пріоритетності. Оскільки рішення щодо розподілу допомоги не можуть ґрунтуватися лише на одному параметрі [15], доцільним є використання багатокритеріального аналізу прийняття рішень. Для соціальної підтримки характерне поєднання критеріїв прямої та оберненої дії, що потребує нормалізації та уніфікації шкал [12]; методично це відповідає формулам (1.2) та (1.3). Окремо важливим є оцінювання інформативності ознак, оскільки надлишкові або слабо пов'язані показники можуть ускладнювати інтерпретацію результатів у системах підтримки прийняття рішень [21].

У прикладних соціальних системах поширеними є лінійні вагові моделі, загальний вигляд яких подано у формулі (1.4). Їх перевагою є інтерпретованість, прозорість і можливість аналізу впливу критеріїв [12]. Експертні правила, бальні класифікації та категоріальні схеми можуть бути корисними, але часто мають нижчу гнучкість і складніше адаптуються до змін соціальної політики [12]. Тому доцільними є моделі з явною структурою критеріїв, відкритою процедурою агрегування, пояснюваністю результатів [11] і підтримкою сценарного аналізу зміни вагових коефіцієнтів [22].

Порівняльний аналіз існуючих інформаційних систем у сфері соціальної підтримки свідчить, що значна частина рішень орієнтована на електронний документообіг, облік звернень або звітність, тоді як аналітична функціональність залишається допоміжною. У частині систем аналітична логіка реалізується безпосередньо в прикладному коді, що ускладнює супровід і знижує рівень формалізації алгоритмів оцінювання [1].

Типовими підходами є транзакційні реєстрові системи, системи статистичного моніторингу, аналітичні DSS та інтегровані платформи гуманітарного менеджменту. Їхніми обмеженнями є недостатня прозорість алгоритмів, складність пояснення індивідуальної пріоритизації та слабка адаптивність до зміни критеріїв оцінювання [17]. За жорсткої фіксації логіки оцінювання в програмному коді будь-яка зміна критеріїв потребує суттєвого втручання у структуру системи [13]. Тому доцільним є відокремлення операційного рівня від аналітичного та використання багат шарової архітектури, у межах якої аналітичні моделі реалізуються у вигляді окремих представлень або процедур обробки даних [12]. Також необхідною є підтримка валідації даних, оскільки недостовірність або неповнота інформації може викривляти результати ранжування [16].

Узагальнене порівняння основних підходів до підтримки прийняття рішень у соціальній сфері наведено в таблиці 1.1.

Як видно з таблиці 1.1, окремі підходи розв'язують лише частину задачі підтримки прийняття рішень. Тому в межах цієї роботи доцільним є поєднання реляційної бази даних, SQL-орієнтованої аналітичної обробки, багатокритеріальної моделі оцінювання та механізмів пояснюваності результатів.

Як бачимо, проведений аналіз предметної області, існуючих підходів до організації даних, багатокритеріального оцінювання та підтримки прийняття рішень дасть змогу конкретизувати проектні завдання подальшої реалізації системи:

- сформуванати реляційну модель даних для зберігання інформації про бенефіціарів, потреби, вразливості та надану допомогу;
- розробити SQL-орієнтований аналітичний конвеєр формування похідних ознак на основі операційних даних;
- забезпечити нормалізацію різнорідних показників для їх коректного порівняння у межах моделі оцінювання;
- побудувати модель багатокритеріального оцінювання пріоритетності бенефіціарів;
- реалізувати механізм пояснюваності результатів через визначення внеску окремих критеріїв у підсумкову оцінку;
- реалізувати серверну частину та інтерфейс введення даних, перегляду аналітичних результатів і таблиці рішень;
- провести експериментальну перевірку коректності, стабільності, чутливості та відтворюваності роботи системи.

Таблиця 1.1 – Порівняння підходів до підтримки прийняття рішень у соціальній сфері

Підхід або тип системи	Переваги	Обмеження	Використано у роботі
Електронні таблиці	Простота використання, швидке внесення даних, доступність для малих організацій	Відсутність надійного контролю цілісності, складність аналітичної обробки, ризик дублювання та помилок	Відмова від фрагментованого зберігання даних
Реєстрові інформаційні системи	Централізований облік бенефіціарів, структуроване зберігання даних, підтримка історії записів	Обмежена аналітична підтримка, недостатня гнучкість оцінювання потреб і пріоритетності	Реляційна база даних PostgreSQL
Системи підтримки прийняття рішень	Формалізація критеріїв, підтримка ранжування альтернатив, можливість обґрунтування рішень	Потреба у прозорих критеріях, виборі ваг і перевірці стійкості результатів	MCDA-модель оцінювання пріоритетності
Інтелектуальні системи та аналітичні панелі	Зручна візуалізація показників, агрегована звітність, підтримка аналізу результатів	Не завжди пояснюють причини індивідуальної пріоритизації та логіку формування рішення	Аналітична панель і пояснення внеску критеріїв

## 1.6 Постановка завдань на кваліфікаційну роботу бакалавра

Ефективність сучасних інформаційно-аналітичних систем у секторі громадських організацій на пряму залежить від детального аналізу оперативних даних, виявлення прихованих соціально-економічних потреб бенефіціарів та правильного вибору моделі оцінювання. Проектування комплексу пріоритезації для розподілу обмежених ресурсів вимагає синергії між надійними алгоритмами нормалізації показників та гнучким програмним забезпеченням, що дозволяє забезпечити стабільне його функціонування в умовах динамічних експлуатаційних навантажень і гарантувати прозорість та відтворюваність прийнятих рішень.

Для досягнення поставленої мети та розв'язання описаної науково-технічної проблеми необхідно виконати такі завдання:

- проаналізувати предметну область, існуючі підходи до управління даними та систем підтримки прийняття рішень у діяльності громадських організацій;
- визначити функціональні й нефункціональні вимоги до проектованої інформаційної системи;
- спроектувати архітектуру системи, реляційну модель даних і структуру бази даних;
- розробити SQL-орієнтований аналітичний конвеєр формування та нормалізації ознак;
- реалізувати модель багатокритеріального оцінювання, ранжування та пояснення пріоритетності бенефіціарів;
- реалізувати серверні компоненти (backend) та користувацький інтерфейс для введення даних і перегляду результатів;
- провести експериментальну перевірку коректності, стабільності, чутливості та відтворюваності роботи системи.

## РОЗДІЛ 2

### ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ ТА МЕТОДІВ РЕАЛІЗАЦІЇ

#### 2.1 Загальна логіка проектування системи

Проектування інформаційної системи виконується як розроблення прикладного ІТ-артефакту, призначеного для підтримки збору, структуризації та аналітичної обробки даних громадської організації. Основним результатом роботи є цілісна інформаційно-аналітична система, що поєднує реляційну модель даних, SQL-орієнтовані аналітичні перетворення, детерміновану модель підтримки прийняття рішень та засоби представлення результатів користувачу.

Загальна логіка проектування відповідає проектному підходу *Design Science Research*, оскільки робота спрямована на створення та перевірку функціонального системного артефакту для розв'язання практичної проблеми. У межах цієї кваліфікаційної роботи DSR використовується не як окрема теоретична методологія, а як послідовність етапів: визначення проблеми, формування вимог, проектування архітектури, побудова моделі даних, розроблення аналітичного конвеєра, формалізація моделі прийняття рішень та перевірка коректності отриманих результатів.

Запропоновану систему можна подати у вигляді кортежу (2.1):

$$I = \langle D, F, N, M, R, V \rangle, \quad (2.1)$$

де: D – рівень збору та зберігання операційних даних про бенефіціарів і пов'язані записи;

F – функція формування аналітичних ознак на основі операційних даних;

N – оператор нормалізації ознак для приведення різнорідних показників до єдиного масштабу;

M – модель підтримки прийняття рішень на основі багатокритеріального аналізу;

R – функція оцінювання, ранжування та класифікації бенефіціарів;

V – підсистема перевірки коректності даних, аналітичних перетворень та результатів оцінювання.

Така формалізація відображає перехід від первинних даних до аналітичних ознак, нормалізованих показників, інтегральної оцінки пріоритетності та перевірки результатів. Проектування системи передбачає розв'язання п'яти основних задач: перетворення операційних даних на аналітичні ознаки, нормалізацію показників, побудову MCDA-моделі пріоритизації, забезпечення пояснюваності результатів і відтворюваності аналітичного конвеєра.

На відміну від систем штучного інтелекту або машинного навчання, запропонована система не виконує прогнозування та не використовує адаптивні моделі. Її логіка базується на реляційній моделі даних, SQL-аналітиці, багатокритеріальному оцінюванні та детермінованих правилах обчислення пріоритетності. Це дозволяє перейти від аналізу предметної області до структурованої моделі системи, придатної для подальшої програмної реалізації та експериментальної оцінки.

## **2.2 Аналіз вимог до системи**

Аналіз вимог визначає функціональні та нефункціональні характеристики інформаційної системи, необхідні для підтримки збору, зберігання, аналітичної обробки даних і формування оцінок пріоритетності бенефіціарів.

До функціональних вимог проектованої системи належать:

- реєстрація бенефіціарів як об'єктів подальшого аналітичного оцінювання;
- фіксація випадків вразливості із зазначенням рівнів серйозності;
- фіксація виявлених потреб із визначенням рівня терміновості;
- облік та відстеження наданої допомоги в динаміці;
- формування похідних аналітичних ознак на основі операційних даних;

- нормалізація ознак для забезпечення їх порівнянності;
- обчислення інтегральної оцінки пріоритетності з використанням моделі багатокритеріального аналізу рішень (MCDA);
- класифікація бенефіціарів за рівнями пріоритетності;
- представлення аналітичних результатів у зрозумілій для користувача формі.

Основними нефункціональними вимогами є:

- відтворюваність – однакові вхідні дані повинні приводити до однакових результатів обчислення пріоритетності;
- прозорість – логіка формування ознак, нормалізації та оцінювання має бути явно задана через аналітичні перетворення;
- цілісність даних – структура бази даних повинна забезпечувати коректні зв'язки між бенефіціарами, записами вразливості, потребами та історією допомоги;
- підтримуваність – аналітична логіка повинна бути розділена на окремі рівні та представлення, що спрощує її перевірку й подальше розширення;
- пояснюваність – користувач повинен мати можливість зрозуміти, які ознаки вплинули на підсумкову оцінку пріоритетності;
- практичність використання – результати аналізу мають подаватися у формі, придатній для використання в робочих процесах громадської організації.

Зазначені вимоги визначають логіку подальшого проектування бази даних, аналітичних представлень, моделі оцінювання та прикладного рівня системи. Технічні деталі реалізації інтерфейсу користувача та середовища розгортання розглядаються в наступних розділах роботи.

### **2.3 Архітектура інформаційно-аналітичної системи**

Архітектура проектованої системи побудована за багаторівневим принципом, що дозволяє відокремити операційне зберігання даних, аналітичні

перетворення, модель підтримки прийняття рішень та подання результатів користувачу.

У структурі системи виділено чотири основні рівні. Рівень операційних даних містить реляційні таблиці для зберігання інформації про бенефіціарів, випадки вразливості, виявлені потреби та надану допомогу. Аналітичний рівень перетворює необроблені записи на структуровані ознаки за допомогою SQL-орієнтованих операцій агрегації. Рівень підтримки прийняття рішень виконує нормалізацію ознак, зважене агрегування та обчислення інтегральної оцінки пріоритетності. Рівень представлення результатів забезпечує доступ користувача до даних, підсумкових оцінок, категорій пріоритетності та пояснень.

Взаємодія компонентів організована як послідовний потік даних. Користувач взаємодіє із системою через рівень представлення, прикладний рівень передає запити до реляційної бази даних, а основні аналітичні перетворення виконуються через SQL-представлення. Результати повертаються до прикладного рівня та відображаються користувачу у вигляді оцінок, категорій і пояснень.

Узагальнено потік даних у системі можна подати як послідовність таких етапів:

- 1) введення та зберігання операційних даних;
- 2) агрегування записів і формування аналітичних ознак;
- 3) нормалізація ознак для забезпечення порівнянності;
- 4) обчислення інтегральної оцінки пріоритетності;
- 5) класифікація результатів за рівнями пріоритету;
- 6) подання результатів і пояснень користувачу.

Узагальнену архітектуру проєктованої системи наведено на рисунку 2.1.

Таким чином, архітектура системи забезпечує детермінований перехід від необроблених даних до аналітичного результату, а кожний етап обробки має окреме призначення та може бути перевірений.

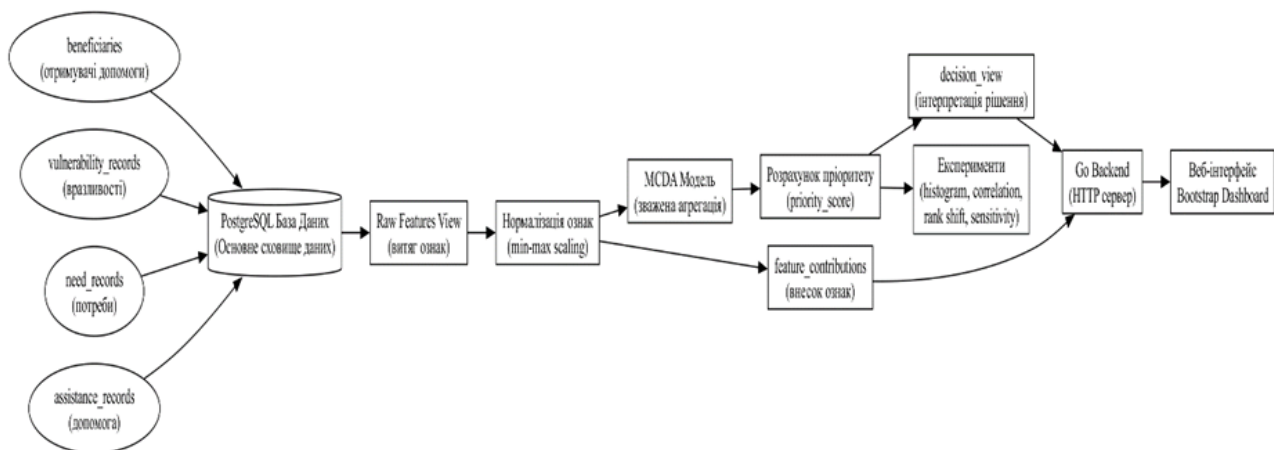


Рисунок 2.1 – Архітектура інформаційно-аналітичної системи підтримки прийняття рішень

## 2.4 Проектування моделі даних

Модель даних проектованої системи визначає структуру зберігання первинної інформації про бенефіціарів, їхні потреби, випадки вразливості та історію надання допомоги. Вона є основою для подальшого формування аналітичних ознак, нормалізації показників і обчислення оцінки пріоритетності.

Концептуальна модель складається з чотирьох основних сутностей: «бенефіціар», «запис про вразливість», «запис про потребу» та «запис про допомогу». Центральною сутністю є бенефіціар, оскільки саме він виступає об'єктом подальшого аналітичного оцінювання та прийняття рішень. Відношення між бенефіціаром і пов'язаними записами мають характер «один до багатьох», що дозволяє зберігати не лише поточний стан, а й набір спостережень, необхідних для аналітичної обробки.

Логічну структуру операційного рівня бази даних, що відображає зв'язки між основною таблицею бенефіціарів та пов'язаними записами про потреби, вразливості й надану допомогу, наведено на рисунку 2.2.

Як показано на рисунку 2.2, таблиця `beneficiaries` є центральною сутністю моделі даних. Таблиці `need_records`, `vulnerability_records` та

assistance\_records пов'язані з нею через зовнішній ключ beneficiary\_id, що забезпечує зберігання декількох записів потреб, вразливостей і випадків надання допомоги для одного бенефіціара.

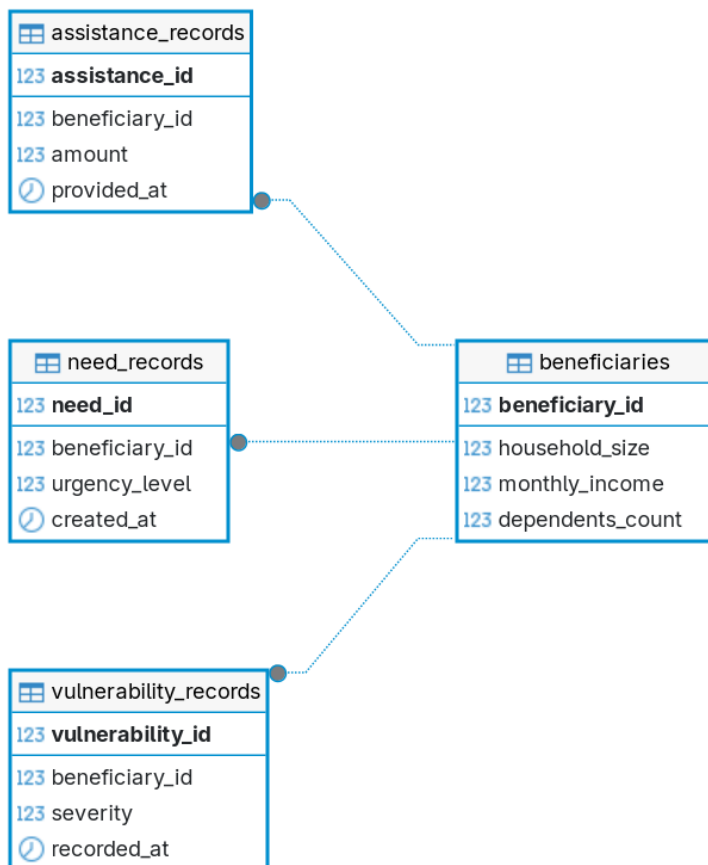


Рисунок 2.2 – Логічна модель операційного рівня бази даних

Формально множину бенефіціарів можна подати як  $V = \{b_1, b_2, \dots, b_n\}$ , а кожного бенефіціара – кортежем  $b_i = (h_i, u_i, d_i)$ , де  $h_i \in \mathbb{N}$  позначає розмір домогосподарства,  $u_i \in \mathbb{R}_{\geq 0}$  – місячний дохід, а  $d_i \in \mathbb{N}$  – кількість утриманців.

Крім того, система визначає три множини спостережувальних подій (2.2):

$$\begin{aligned}
 V &= \{(b_i, s, t)\} \text{ – «спостереження щодо вразливостей»}; \\
 N &= \{(b_i, u, t)\} \text{ – «спостереження щодо потреб»}; \\
 A &= \{(b_i, a, t)\} \text{ – «записи про надану допомогу»},
 \end{aligned}
 \tag{2.2}$$

де  $s, u \in \{1, 2, 3, 4, 5\}$  – позначають порядкові шкали серйозності та терміновості, а  $a \in \mathbb{R}_{\geq 0}$  позначає обсяг виділених ресурсів.

Логічна модель реалізує ці сутності у вигляді взаємопов'язаних реляційних таблиць. Основна таблиця `beneficiaries` містить базові соціально-економічні характеристики: `beneficiary_id`, `household_size`, `monthly_income` та `dependents_count`. Пов'язані записи про вразливість, потреби та допомогу зберігаються в окремих таблицях, пов'язаних із таблицею бенефіціарів за допомогою зовнішніх ключів.

Операційний рівень системи можна подати як сукупність множин  $O = \{B, V, N, A\}$ , що зберігають необроблені спостережувальні кортежі. Первинні ключі використовуються для однозначної ідентифікації записів, зовнішні ключі забезпечують референтну цілісність, а обмеження допустимих значень зменшують ризик некоректного введення даних.

Фізична модель реалізується в середовищі PostgreSQL. На цьому рівні визначаються таблиці, типи даних, ключі, обмеження цілісності та індекси, необхідні для ефективного виконання операцій зберігання й аналітичної вибірки даних [13]. Таким чином, модель даних забезпечує структуроване зберігання первинних записів і підготовку даних до подальшого аналітичного конвеєра.

## 2.5 Проектування аналітичного конвеєра

Аналітичний конвеєр системи забезпечує послідовне перетворення операційних даних на структуровані ознаки, нормалізовані показники та результати, придатні для подальшого обчислення пріоритетності бенефіціарів. Узагальнено його роботу можна подати як перехід від зчитування операційних даних до формування ознак, нормалізації, обчислення підсумкових оцінок, категорій пріоритету та пояснень.

Аналітична система визначає набір похідних показників, які відображають соціально-економічні характеристики та рівень вразливості бенефіціарів. До таких показників належать економічний потенціал, рівень залежності, структурна вразливість, інтенсивність потреб та сукупний обсяг попередньо отриманої допомоги.

Формально система визначає відповідність  $F : B \cup V \cup N \cup A \rightarrow R^5$ , яка перетворює операційні дані на п'ятивимірний вектор аналітичних ознак. Кожен бенефіціар перетворюється на вектор (2.3):

$$X_i = (X_{i1}, X_{i2}, X_{i3}, X_{i4}, X_{i5}), \quad (2.3)$$

де  $x_{i1}$  – економічний потенціал;

$x_{i2}$  – рівень залежності;

$x_{i3}$  – структурна вразливість;

$x_{i4}$  – інтенсивність потреб;

$x_{i5}$  – сукупний обсяг попередньо отриманої допомоги.

Компоненти вектора аналітичних ознак (2.3) визначаються з (2.4):

$$\begin{aligned} x_{i1} &= y_i / h_i; \quad x_{i2} = (d_i + 1) / h_i; \\ x_{i3} &= (1 / |V_i|) \times \sum_{(b_i, s, t) \in V_i} S; \quad x_{i4} = (1 / |N_i|) \times \sum_{(b_i, u, t) \in N_i} u; \\ x_{i5} &= \sum_{(b_i, a, t) \in A_i} a. \end{aligned} \quad (2.4)$$

У формулі (2.4) перша ознака описує дохід у розрахунку на одного члена домогосподарства, друга – демографічне навантаження, третя – середній рівень серйозності зафіксованих випадків вразливості, четверта – середній рівень терміновості зареєстрованих потреб, а п'ята – накопичений обсяг уже наданої допомоги. Якщо для певного бенефіціара відсутні записи у відповідній множині спостережень, система призначає нульове значення  $x_{ij} = 0$ , що дозволяє сформулювати повний вектор ознак для подальших етапів обробки.

Оскільки аналітичні ознаки мають різні шкали вимірювання, перед обчисленням інтегральної оцінки вони приводяться до спільного інтервалу. Для цього використовується min-max нормалізація, загальний вигляд якої наведено у формулі (1.2), а для критеріїв оберненої дії – у формулі (1.3).

Для кожної ознаки визначаються мінімальне та максимальне значення:  $x_j^{\min} = \min_i x_{ij}$  та  $x_j^{\max} = \max_i x_{ij}$ . Тоді нормалізований вектор ознак визначають з (2.5):

$$\begin{aligned} p_{i1} &= 1 - [(x_{i1} - x_1^{\min}) / (x_1^{\max} - x_1^{\min})]; & p_{i2} &= (x_{i2} - x_2^{\min}) / (x_2^{\max} - x_2^{\min}); \\ p_{i3} &= (x_{i3} - x_3^{\min}) / (x_3^{\max} - x_3^{\min}); & p_{i4} &= (x_{i4} - x_4^{\min}) / (x_4^{\max} - x_4^{\min}); \\ p_{i5} &= 1 - [(x_{i5} - x_5^{\min}) / (x_5^{\max} - x_5^{\min})]. \end{aligned} \quad (2.5)$$

Отже, для кожного бенефіціара формується нормалізований вектор  $p_i \in [0, 1]^5$ . Як показано у формулі (2.5), ознаки  $x_{i1}$  та  $x_{i5}$  нормалізуються в інверсній формі, оскільки більший дохід на одного члена домогосподарства або більший обсяг уже отриманої допомоги зменшує рівень пріоритетності.

Аналітичний конвеєр реалізується як послідовність реляційних представлень: `beneficiary_features_raw`, `beneficiary_features_normalized`, `beneficiary_priority_scores` та `beneficiary_decision_view`. Така структура забезпечує послідовний перехід від операційних даних до нормалізованих ознак, оцінки пріоритетності, категорії рішення та пояснюваного результату.

Структуру SQL-представлень аналітичного конвеєра наведено на рисунку 2.3.

## 2.6 Проектування моделі підтримки прийняття рішень

Модель підтримки прийняття рішень призначена для перетворення нормалізованих аналітичних ознак на інтегральну оцінку пріоритетності бенефіціарів. У системі використовується підхід багатокритеріального аналізу

рішень (MCDA), який забезпечує формалізоване, відтворюване та пояснюване оцінювання.

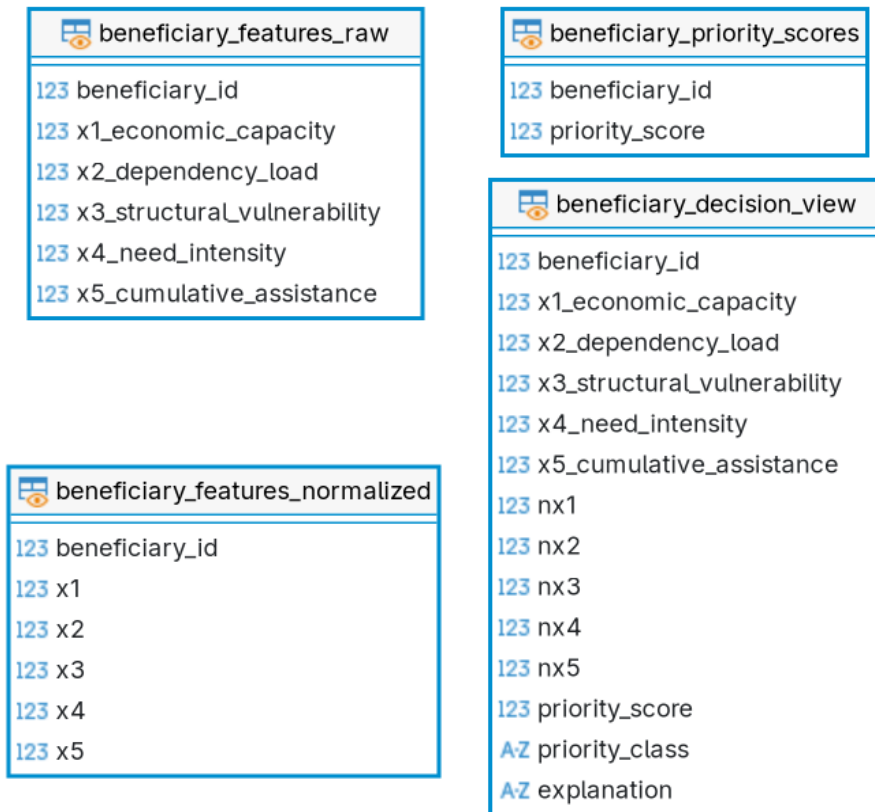


Рисунок 2.3 – Структура SQL-представлень аналітичного конвеєра

Формально модель підтримки прийняття рішень можна розглядати як функцію  $S : R^5 \rightarrow [0, 1]$ . Для кожного бенефіціара інтегральна оцінка пріоритетності обчислюється як зважена адитивна функція (2.6) [18]:

$$S_i = \sum w_j \cdot n_{ij}, \quad (2.6)$$

де  $S_i$  – інтегральна оцінка пріоритетності бенефіціара  $i$ ;

$w_j$  – ваговий коефіцієнт  $j$ -ї ознаки;

$n_{ij}$  – нормалізоване значення  $j$ -ї ознаки для бенефіціара  $i$ .

Формула (2.6) є основою моделі підтримки прийняття рішень, оскільки саме за її результатом виконується ранжування бенефіціарів і подальша

класифікація рівня пріоритетності. У базовій конфігурації системи використовується рівномірний розподіл ваг (2.7):

$$w_j=0,2, \quad \sum w_j=1. \quad (2.7)$$

Умова (2.7) визначає базову політику оцінювання, за якої жодна з п'яти ознак не має початкової переваги над іншими. За умови рівномірного розподілу ваг (2.7) модель (2.6) набуває вигляду (2.8):

$$S(n_i)=0,2\sum n_{ij}. \quad (2.8)$$

Отримані значення  $S_i$  використовуються для ранжування бенефіціарів за ступенем пріоритетності та підтримки рішень щодо розподілу ресурсів.

Для підтримки практичного прийняття рішень безперервна оцінка пріоритетності перетворюється на дискретні категорії (2.9):

$$C(S_i)= \begin{cases} \text{«ВИСОКИЙ ПРІОРИТЕТ»}, S_i \geq 0,95; \\ \text{«СЕРЕДНІЙ ПРІОРИТЕТ»}, 0,60 \leq S_i < 0,95; \\ \text{«НИЗЬКИЙ ПРІОРИТЕТ»}, S_i < 0,60. \end{cases} \quad (2.9)$$

Згідно з формулою (2.9), кожен бенефіціар отримує одну з трьох категорій пріоритетності залежно від значення інтегральної оцінки  $S_i$ , обчисленої за формулою (2.7). У результаті формується поділ множини бенефіціарів  $B = B_H \cup B_M \cup B_L$  ( $B_H$ ,  $B_M$  та  $B_L$  – відповідають групам високого, середнього й низького пріоритету).

## 2.7 Пояснюваність та простежуваність результатів

Пояснюваність результатів забезпечується лінійною структурою MCDA-моделі, обчисленням внесків окремих ознак і простежуваністю переходу від первинних записів до підсумкової категорії пріоритетності.

Оскільки інтегральна оцінка пріоритетності формується як зважена сума нормалізованих ознак, внесок кожної ознаки визначають з (2.10):

$$C_{ij}=w_j \cdot n_{ij}, \quad (2.10)$$

де  $C_{ij}$  – внесок  $j$ -ї ознаки у підсумкову оцінку бенефіціара  $i$ ;

$w_j$  – вага ознаки;

$n_{ij}$  – нормалізоване значення  $j$ -ї ознаки для бенефіціара  $i$ .

Формула (2.10) дозволяє розкласти інтегральну оцінку (2.7) на окремі складові та визначити, які характеристики найбільше вплинули на пріоритетність. Для формування пояснення система використовує правило виявлення домінантних ознак (2.11):

$$E_i = \sum I(n_{ij} > \tau) \cdot \phi_j, \quad (2.11)$$

де  $\tau=0,7$ ;

$I$  – індикаторна функція;

$\phi_j$  – семантичний опис ознаки, що має суттєвий вплив на результат [12].

Згідно з формулою (2.11), пояснення формується лише для тих ознак, нормалізоване значення яких перевищує заданий поріг  $\tau$ . Семантичні дескриптори визначаються так:

$\phi_1$  – «Низький економічний потенціал»;

$\phi_2$  – «Високе демографічне навантаження»;

$\phi_3$  – «Висока структурна вразливість»;

$\phi_4$  – «Висока інтенсивність потреб»;

$\phi_5$  – «Незначний обсяг попередньо отриманої допомоги».

Простежуваність результатів забезпечується послідовною структурою аналітичного конвеєра: підсумкова оцінка формується на основі

нормалізованих ознак, нормалізовані ознаки – на основі агрегованих показників, а агреговані показники – на основі первинних записів про бенефіціарів, вразливості, потреби та надану допомогу. У базі даних підсумковий інтерпретаційний шар реалізується через представлення `beneficiary_decision_view`, яке поєднує числову оцінку, категорію рішення та семантичне пояснення домінуючих факторів.

## 2.8 Валідація проектних рішень

Валідація проектних рішень спрямована на перевірку коректності структури даних, аналітичних перетворень, нормалізації ознак, обчислення інтегральної оцінки пріоритетності та відтворюваності результатів. У межах цього розділу валідація розглядається як перевірка правильності проектної логіки системи, тоді як детальний аналіз розподілу оцінок, стабільності ранжування та чутливості до зміни ваг подано нижче.

Технічна перевірка передбачає контроль коректності реляційної схеми, зв'язків між таблицями, обмежень цілісності та виконання SQL-запитів. Функціональна перевірка охоплює послідовність основних етапів системи: збереження операційних даних, формування аналітичних ознак, нормалізацію показників, обчислення інтегральної оцінки, класифікацію результатів та формування пояснення.

Коректність аналітичного конвеєра перевіряється через набір формальних обмежень, які повинні виконуватися для всіх бенефіціарів і всіх ознак. По-перше, нормалізовані значення ознак повинні належати інтервалу  $[0,1]$  (2.12):

$$\forall i, j : 0 \leq n_{ij} \leq 1. \quad (2.12)$$

Обмеження (2.12) підтверджує коректність етапу нормалізації та забезпечує порівнянність усіх ознак у межах MCDA-моделі. По-друге,

інтегральна оцінка пріоритетності також повинна залишатися в межах допустимого інтервалу  $[0,1]$  (2.13):

$$0 \leq S_i \leq 1. \quad (2.13)$$

Оскільки оцінка формується за формулою (2.7), виконання обмеження (2.13) підтверджує узгодженість нормалізації, вагових коефіцієнтів і моделі агрегування. По-третє, для кожного бенефіціара має бути сформований результат оцінювання, тобто  $\forall i : S_i \neq \emptyset$ .

Окремо враховується випадок структурної виродженості ознаки, коли мінімальне та максимальне значення збігаються. У такій ситуації застосовується правило (2.14):

$$x_j^{\max} = x_j^{\min} \Rightarrow n_{ij} = 0. \quad (2.14)$$

Правило (2.14) дозволяє уникнути ділення на нуль під час нормалізації та забезпечує стабільність системи за умови вироджених розподілів ознак.

Відтворюваність результатів забезпечується детермінованою структурою аналітичного конвеєра та послідовним виконанням SQL-перетворень. За однакових вхідних даних система повинна формувати однакові аналітичні ознаки, нормалізовані значення, інтегральні оцінки та категорії пріоритетності. У межах проектування використано синтетичні дані, що зменшує ризики обробки реальної персональної інформації; у разі практичного впровадження необхідно передбачити контроль доступу, захист персональних даних і перевірку впливу вагової структури на результати пріоритизації. Крім того, для систем підтримки прийняття рішень доцільно враховувати загрози, пов'язані з несанкціонованим втручанням, викривленням вхідних даних і некоректною інтерпретацією результатів оцінювання [23].

Узагальнення відповідності основних вимог до системи, реалізованих компонентів та способів їх перевірки наведено в таблиці 2.1.

Таблиця 2.1 – Відповідність вимог, реалізації та способів перевірки проєктованої системи

Вимога до системи	Реалізація	Спосіб перевірки
Структуроване зберігання даних	Операційні таблиці PostgreSQL для бенефіціарів, потреб, вразливостей і наданої допомоги	Перевірка первинних і зовнішніх ключів, обмежень цілісності та допустимих діапазонів значень
Формування аналітичних ознак	SQL-представлення для агрегування операційних записів у похідні аналітичні показники	Контроль результатів агрегації, перевірка кількості сформованих записів і відсутності критичних пропусків
Нормалізація показників	Аналітичні SQL-представлення з прямою та інверсною min-max нормалізацією	Перевірка належності нормалізованих значень до інтервалу [0, 1] та контроль вироджених ознак
Обчислення пріоритетності та ранжування	SQL-представлення для розрахунку інтегральної оцінки <code>priority_score</code> і впорядкування бенефіціарів	Аналіз порядку ранжування, перевірка меж підсумкової оцінки та порівняння результатів за різних сценаріїв
Пояснюваність результатів	Представлення внесків окремих критеріїв і формування інтерпретації рішення	Перегляд внесків ознак у підсумкову оцінку та перевірка узгодженості пояснення з домінуючими факторами
Відтворюваність аналітичного конвеєра	Послідовний запуск SQL-скриптів і перевірочних запитів у контрольованому середовищі	Повторний запуск конвеєра з однаковими вхідними даними та порівняння отриманих результатів

Як видно з таблиці 2.1, кожна ключова вимога до проєктованої системи має відповідний механізм реалізації та спосіб перевірки. Це забезпечує зв'язок між етапами проєктування, реалізації та подальшої експериментальної оцінки системи.

Після формалізації архітектури, моделі даних, аналітичного конвеєра та правил валідації доцільно перейти до експериментальної перевірки, яка показує, як спроектовані рішення працюють на контрольному наборі даних та чи відповідають визначеним обмеженням коректності.

## РОЗДІЛ 3

### ПРАКТИЧНА РЕАЛІЗАЦІЯ

#### 3.1 Технологічне середовище реалізації

Система реалізована як інформаційно-аналітичне середовище, у якому основна логіка зберігання, перетворення та аналітичної обробки даних зосереджена на рівні бази даних. Такий підхід забезпечує прозорість обчислень, відтворюваність результатів і придатність рішення для використання в умовах обмежених технічних ресурсів громадської організації.

Основою реалізації є PostgreSQL, який використовується не лише як реляційна система керування базами даних, а й як аналітичний рівень. На рівні PostgreSQL виконуються зберігання операційних записів, формування аналітичних представлень, нормалізація показників, обчислення оцінок пріоритетності та підготовка результатів для інтерфейсу користувача. Нормалізація та обчислення інтегральної оцінки реалізують проектні правила, визначені відповідно у формулах (2.7) та (2.8). Для цього використовуються можливості SQL, зокрема загальні табличні вирази, віконні функції, агрегатні функції та подання. Відповідна логіка реалізована у файлах `decision_schema_layer.sql`, `scores.sql` та `decision_view.sql`.

Backend-рівень реалізовано мовою Go у файлі `ui/main.go`. Він відповідає за запуск HTTP-сервера, маршрутизацію запитів, підключення до PostgreSQL, виконання SQL-запитів і передавання результатів до HTML-шаблонів. Основні аналітичні обчислення не дублюються в Go-коді, оскільки вони винесені на рівень SQL-представлень; backend виконує роль проміжного сервісного рівня між базою даних та інтерфейсом користувача [24].

Користувацький інтерфейс побудовано на основі серверних HTML-шаблонів з каталогу `templates/`. Шаблони `analytics.html`, `profile.html` та `forms.html` використовуються для відображення аналітичних результатів, профілю бенефіціара та форм введення даних. Стилізація інтерфейсу

здійснюється за допомогою `Bootstrap`, а `JavaScript` використовується лише для допоміжної інтерактивності, наприклад фільтрації таблиць.

Для розгортання використовується контейнеризоване середовище `PostgreSQL` і скрипт `pipeline.sh`. Він послідовно ініціалізує схему бази даних, створює аналітичні представлення, генерує синтетичні дані та виконує перевірочні й експериментальні `SQL`-запити. `Docker` забезпечує ізоляцію середовища та повторюваність запуску системи [24].

Отже, технологічне середовище системи побудоване навколо `PostgreSQL`, `SQL`-аналітики, `backend`-рівня на `Go`, серверних `HTML`-шаблонів і контейнеризованого запуску. Така структура підтримує детерміновану обробку даних і спрощує перевірку роботи системи.

### 3.2 Реалізація бази даних та аналітичного рівня

Операційна схема визначає транзакційну структуру системи та реалізована у файлі `operational_schema_layer.sql`. Вона містить чотири основні реляційні сутності: `beneficiaries`, `vulnerability_records`, `need_records` та `assistance_records`. Повні `SQL`-лістинги створення операційного шару бази даних, аналітичних представлень, обчислення оцінок пріоритетності та експериментальних перевірок наведено в додатку А.

Таблиця `beneficiaries` є центральною сутністю операційної схеми: кожен її запис відповідає окремому бенефіціару або домогосподарству, для якого надалі формується аналітичний профіль і оцінка пріоритетності. У цій таблиці зберігаються базові соціально-економічні атрибути: розмір домогосподарства, щомісячний дохід і кількість утриманців.

Таблиці `vulnerability_records`, `need_records` та `assistance_records` зберігають динамічні записи про вразливість, потреби та надану допомогу. Зв'язки між таблицями забезпечуються зовнішніми ключами, які посилаються на ідентифікатор бенефіціара та підтримують

референтну цілісність. Індокси застосовуються до зовнішніх ключів і часових атрибутів, що спрощує виконання агрегуючих запитів на аналітичному рівні.

Аналітичний рівень бази даних реалізовано у файлі `decision_schema_layer.sql` як послідовність SQL-представлень, що перетворюють операційні записи на структуровані показники для підтримки прийняття рішень. Цей рівень відокремлює первинні транзакційні дані від аналітичних ознак і забезпечує перехід від необроблених записів до нормалізованих змінних.

Перший етап реалізовано через представлення `beneficiary_features_raw`. Воно агрегує записи з операційних таблиць і формує початкові аналітичні ознаки для кожного бенефіціара: економічний потенціал, рівень утриманства, структурну вразливість, інтенсивність потреб і сукупний обсяг раніше отриманої допомоги. Для цього використовуються агрегатні SQL-функції AVG, SUM та COUNT. Структура цих ознак відповідає вектору (2.3) та його компонентам (2.4).

Представлення `feature_statistics` обчислює мінімальні та максимальні значення для кожної ознаки в межах поточного набору даних. Ці параметри використовуються у представленні `beneficiary_features_normalized`, де ознаки приводяться до шкали відповідно до формули (2.5). Для вироджених випадків, коли мінімальне та максимальне значення збігаються, SQL-логіка використовує обробку значень NULL, що реалізує правило (2.14) і запобігає діленню на нуль.

Під час нормалізації враховується напрям впливу ознак. Структурна вразливість, рівень утриманства й інтенсивність потреб нормалізуються прямо пропорційно, оскільки їх зростання підвищує пріоритетність. Економічний потенціал і сукупний обсяг раніше отриманої допомоги нормалізуються обернено пропорційно, оскільки їх зростання зменшує потребу в пріоритетній підтримці.

Отже, представлення `beneficiary_features_raw`, `feature_statistics` та `beneficiary_features_normalized` утворюють основу аналітичного рівня системи. Вони забезпечують перехід від операційної

бази даних до структурованого аналітичного профілю бенефіціара, який надалі використовується для обчислення інтегральної оцінки та формування пояснюваного результату.

Узагальнення технологічного стеку реалізованої інформаційно-аналітичної системи наведено в таблиці 3.1.

Таблиця 3.1 – Технологічний стек інформаційно-аналітичної системи

Компонент	Технологія	Призначення
Система керування базами даних	PostgreSQL	Зберігання операційних даних, реалізація SQL-аналітики, формування аналітичних представлень і результатів оцінювання
Backend-рівень	Go	Обробка HTTP-запитів, взаємодія з базою даних, передавання результатів аналітичної обробки до інтерфейсу користувача
Frontend-рівень	HTML, CSS, Bootstrap	Формування користувацького інтерфейсу для введення даних, перегляду аналітичної панелі, таблиці рішень і профілю бенефіціара
Контейнеризоване середовище	Docker	Забезпечення відтворюваного запуску системи та узгодженого середовища виконання компонентів
SQL-скрипти	Файли .sql	Створення структури бази даних, аналітичних представлень, розрахунок інтегральної оцінки та виконання перевірочних запитів

Як видно з таблиці 3.1, технологічний стек системи поєднує PostgreSQL, Go, HTML/CSS/Bootstrap, Docker і SQL-скрипти. Такий набір засобів забезпечує структуроване зберігання даних, відтворювану SQL-аналітику, просту серверну взаємодію та зрозуміле подання результатів користувачу.

Рівень підтримки прийняття рішень реалізований через представлення `beneficiary_features_normalized`, `beneficiary_priority_scores`, `feature_contributions` та `beneficiary_decision_view`. Вони забезпечують послідовний перехід від нормалізованих ознак до інтегральної оцінки пріоритетності, категорії рішення та пояснюваного результату.

Представлення `beneficiary_priority_scores` реалізує обчислення інтегральної оцінки за MCDA-моделлю, визначеною у формулі (2.7). Ваги критеріїв задані безпосередньо в SQL-логіці; у базовій конфігурації використовується рівновагова схема, наведена у формулі (2.8). Категоріальна

інтерпретація результату формується у представленні `beneficiary_decision_view` відповідно до правила класифікації (2.9).

Для забезпечення пояснюваності використовується представлення `feature_contributions`, у якому обчислюється внесок кожної ознаки в підсумкову оцінку відповідно до формули (2.10). Фінальне представлення `beneficiary_decision_view` поєднує початкові ознаки, нормалізовані значення, інтегральний бал, клас пріоритетності та текстові пояснення, сформовані на основі правила (2.11). Саме це представлення використовується як основне джерело даних для `backend`-рівня та інтерфейсу користувача.

### **3.3 Інтерфейс користувача для введення даних та подання результатів**

Інтерфейс користувача є прикладним рівнем реалізованої системи, який забезпечує введення даних і перегляд аналітичних результатів. Для громадської організації він має бути зрозумілим для нетехнічного персоналу, підтримувати структуроване внесення інформації та подавати результати оцінювання у практично придатній формі.

Інтерфейс виконує дві основні функції: введення даних про бенефіціарів і представлення результатів багатокритеріального оцінювання. Веб-інтерфейс реалізовано за допомогою серверних HTML-шаблонів, які формують сторінки введення даних, аналітичної панелі, таблиці рішень і профілю бенефіціара. Навігація підтримує типовий робочий процес: внести або уточнити дані, переглянути результати, проаналізувати таблицю рішень і за потреби відкрити профіль окремого бенефіціара.

Основний механізм збору даних реалізовано за допомогою структурованих форм. Форма реєстрації бенефіціара містить чисельність домогосподарства, щомісячний дохід і кількість утриманців. Ці показники використовуються під час формування аналітичних ознак і розрахунку оцінки пріоритетності в моделі MCDA.

На рівні інтерфейсу передбачено базову перевірку введених даних: обов'язковість заповнення ключових полів, мінімальні допустимі значення та числові типи введення. Окрім базових характеристик, система підтримує внесення пов'язаних записів про вразливість, потреби та раніше надану допомогу. Кожен тип запису відповідає окремому аспекту подальшого оцінювання. Приклад сторінки введення даних наведено на рисунку 3.1.

The screenshot shows a web interface for 'Аналітика соціальної допомоги' (Social Assistance Analytics). At the top right, there are two buttons: 'Введення даних' (Data Entry) and 'Таблиця рішень' (Decision Table). The main content is divided into two columns:

- Реєстрація нового бенефіціара (Registration of a new beneficiary):** This form contains three input fields: 'Кількість осіб у домогосподарстві' (Number of people in the household), 'Місячний дохід' (Monthly income), and 'Кількість утриманців' (Number of dependents). Each field has a dropdown arrow on the right. Below these fields is a blue button labeled 'Додати бенефіціара' (Add beneficiary).
- Додати записи (Add records):** This form contains three input fields: 'ID Бенефіціара' (Beneficiary ID), 'Тип запису' (Record type) with a dropdown menu showing 'Вразливість (Серйозність 1-5)' (Vulnerability (Seriousness 1-5)), and 'Значення (Серйозність, Терміновість, або Сума)' (Value (Seriousness, Urgency, or Sum)) with a dropdown arrow. Below these fields is a green button labeled 'Додати запис' (Add record).

Рисунок 3.1 – Сторінка введення даних про бенефіціара та пов'язані записи

Як видно з рисунка 3.1, інтерфейс введення даних поділено на форму реєстрації нового бенефіціара та форму додавання пов'язаних записів. Такий поділ узгоджується з реляційною структурою бази даних і дозволяє поступово доповнювати профіль бенефіціара без дублювання основної інформації.

Панель аналітичних результатів призначена для подання результатів оцінювання у формі, зручній для перегляду та інтерпретації користувачем. У системах підтримки прийняття рішень візуалізація виконує інтерпретаційну функцію, оскільки спрощує сприйняття результатів аналітичного оцінювання та підтримує управлінський аналіз.

У реалізованій системі панель відображає кількість бенефіціарів, середній бал пріоритетності, стандартне відхилення, гістограму розподілу оцінок і показники впливу окремих критеріїв на підсумковий бал. Таке подання дозволяє користувачу швидко оцінити загальний стан набору даних, побачити

групи з різним рівнем пріоритетності та перейти до детального аналізу конкретних записів.

Основним елементом інтерфейсу підтримки прийняття рішень є таблиця рішень, у якій для кожного бенефіціара подано ідентифікатор, розрахований бал пріоритетності, категорію пріоритету та текстове пояснення. Така форма дозволяє швидко визначити випадки з високим рівнем пріоритетності та перейти до профілю конкретного бенефіціара.

Категорія пріоритету спрощує інтерпретацію числової оцінки, а поле пояснення пов'язує підсумковий результат із факторами, що вплинули на його формування. Завдяки цьому таблиця відображає не лише числовий результат MCDA-моделі, а й змістову інтерпретацію рішення.

Фрагмент шаблону, що відповідає за відображення підсумкової оцінки, класу пріоритетності та пояснення результату, наведено в лістингу 3.1.

Лістинг 3.1 – Фрагмент HTML-шаблону таблиці рішень

---

```

{{range .Decisions}}
<tr>
<td><a href="/beneficiary?id={{.ID}}">#{{.ID}}</a></td>
<td>{{printf "%.4f" .Score}}</td>
<td><span class="badge">{{.Class}}</span></td>
<td>{{.Explanation}}</td>
</tr>
{{end}}

```

---

Кінець лістингу 3.1

Інтерфейс також підтримує зміну ваг п'яти критеріїв оцінювання: економічного стану, залежності, вразливості, потреби та попередньо наданої допомоги. Після зміни ваг система перераховує результати ранжування відповідно до заданих параметрів MCDA-моделі та показує порівняння базового і змодельованого ранжування. Приклад сторінки розширеної аналітики та моделювання ваг наведено на рисунку 3.2.

Як видно з рисунка 3.2, інтерфейс поєднує статистику, розподіл оцінок, форму налаштування ваг, таблицю зміни рангів та індикатори впливу ознак.

Для роботи з великими наборами записів таблиця доповнена пошуком, фільтрацією та пагінацією; вибрані вагові коефіцієнти зберігаються в параметрах запиту під час переходу між сторінками.

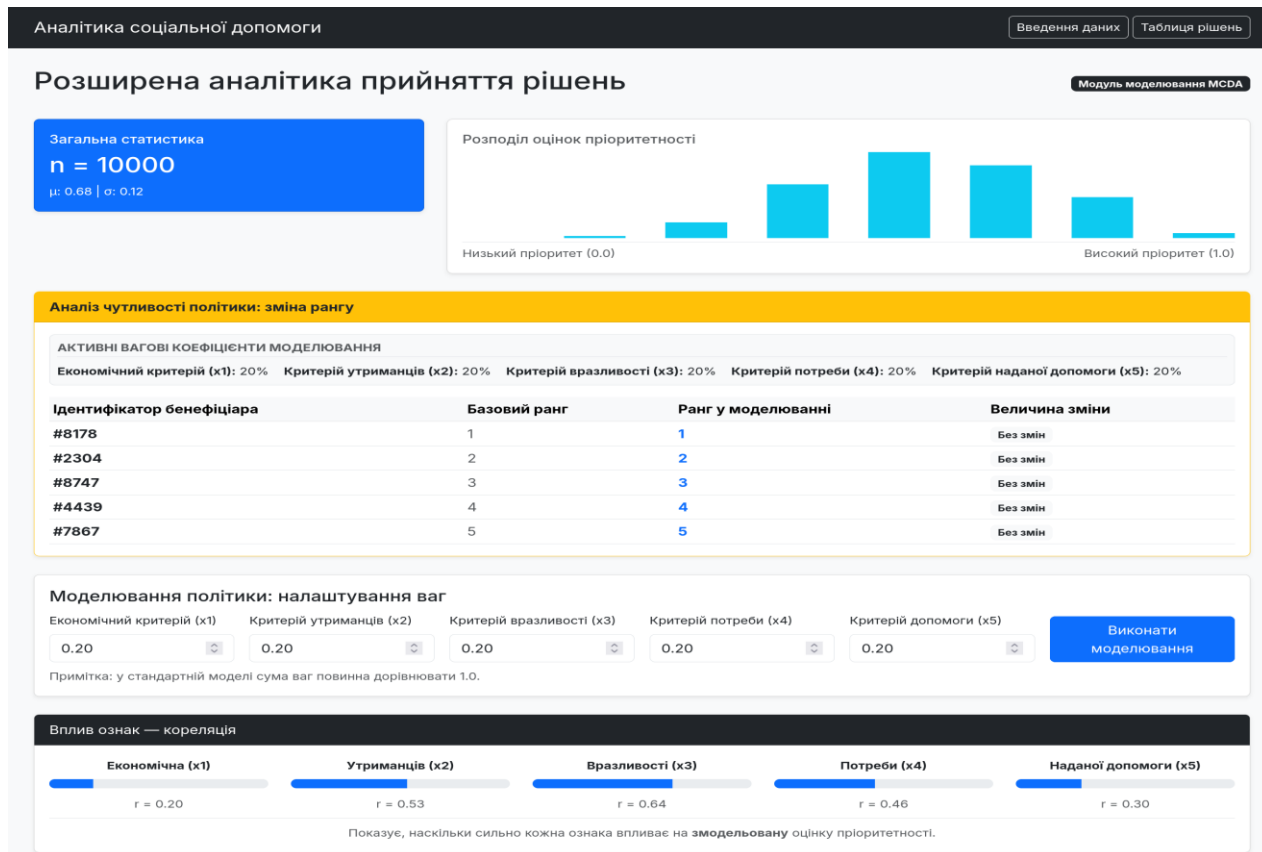


Рисунок 3.2 – Сторінка розширеної аналітики та моделювання ваг MCDA

Система містить спеціальний перегляд профілю окремого бенефіціара, доступний через гіперпосилання в таблиці рішень. Він подає базові соціально-економічні характеристики, історію заявлених потреб, записи вразливості та інформацію про раніше надану допомогу. Такий перегляд дозволяє простежити вихідні дані, які вплинули на оцінку пріоритетності, і підтримує аудит окремого випадку.

Приклад сторінки профілю окремого бенефіціара наведено на рисунку 3.3.

Як видно з рисунка 3.3, профіль поділено на блок базових характеристик та таблиці пов'язаних записів. Це дозволяє користувачу переглянути вхідні дані

та зрозуміти, які записи були використані для формування аналітичного профілю й оцінки пріоритетності.

Бенефіціар #3261	
Кількість осіб у домогосподарств	5
Утриманці	4
Місячний дохід	\$437.45

Історія заявлених потреб	
Дата запису	Рівень терміновості
2026-05-10T13:54:48.580602Z	Рівень 5/5
2026-04-17T13:14:18.743337Z	Рівень 4/5
2026-04-12T14:35:07.703933Z	Рівень 3/5
2026-02-19T09:32:28.204472Z	Рівень 5/5

Історія вразливості	
Дата	Рівень серйозності
2026-04-15T12:51:27.776175Z	4/5
2025-12-11T03:51:06.723361Z	5/5

Історія надання допомоги	
Дата	Сума
2026-04-17T09:13:53.238348Z	\$50.36
2026-02-05T18:23:05.534489Z	\$52.26

Рисунок 3.3 – Сторінка профілю бенефіціара з історією потреб, вразливості та наданої допомоги

### 3.4 Реалізація backend та взаємодії з інтерфейсом користувача

Backend-рівень системи реалізовано мовою Go як легкий веб-додаток, що забезпечує зв'язок між інтерфейсом користувача і базою даних PostgreSQL. Основна логіка розміщена у файлі main.go, де визначено запуск HTTP-сервера, маршрутизацію запитів, підключення до бази даних та виконання SQL-запитів.

Структура backend побудована навколо HTTP-обробників, які приймають запити від користувача, звертаються до відповідних таблиць і SQL-представлень PostgreSQL, отримують результати та передають їх до HTML-шаблонів. Це зберігає чіткий розподіл відповідальності: база даних виконує зберігання й аналітичну обробку, а backend координує доступ до результатів і підготовку їх до відображення.

Як приклад організації серверного рівня у лістингу 3.2 наведено фрагмент маршрутизації HTTP-запитів, що забезпечує доступ до головної сторінки, форм введення даних, аналітичної панелі та профілю бенефіціара.

## Лістинг 3.2 – Фрагмент маршрутизації HTTP-запитів backend-рівня

---

```

http.HandleFunc("/", dashboardHandler)
http.HandleFunc("/add", formsHandler)
http.HandleFunc("/submit-beneficiary", submitBeneficiary)
http.HandleFunc("/submit-record", submitRecord)
http.HandleFunc("/analytics", analyticsHandler)
http.HandleFunc("/beneficiary", beneficiaryProfileHandler)

log.Fatal(http.ListenAndServe(":8080", nil))

```

---

Кінець лістингу 3.2

Аналітичні дані не обчислюються безпосередньо в кодї Go. Backend виконує SQL-запити до підготовлених представлень, що містять аналітичні ознаки, нормалізовані показники, інтегральні оцінки та підсумкові рішення. Форми введення передають отримані дані до операційної схеми бази даних, а аналітична сторінка та профіль бенефіціара відображають результати, сформовані SQL-рівнем. Отже, backend виконує сервісну функцію і не дублює аналітичну логіку PostgreSQL.

### 3.5 Інтеграція, перевірка та запуск системи

Інтеграція компонентів побудована навколо послідовного потоку даних між базою даних, backend-рівнем та інтерфейсом користувача. Дані, введені через інтерфейс, передаються backend-рівнем до операційної схеми PostgreSQL, після чого обробляються аналітичними SQL-представленнями. Далі формуються аналітичні ознаки, нормалізовані показники, інтегральні оцінки пріоритетності та підсумкове представлення рішення, яке повертається в інтерфейс.

Перевірка реалізації виконується за допомогою файлу `validation.sql`. Він містить SQL-запити для контролю меж нормалізації, нульових або відсутніх значень, допустимих діапазонів нормалізованих показників і розподілу підсумкових оцінок. Повні SQL-лістинги створення операційного шару бази даних, аналітичних представлень, обчислення оцінок пріоритетності

та експериментальних перевірок наведено в додатку А. Такі перевірки реалізують формальні обмеження коректності, визначені у формулах (2.12), (2.13) та (2.14).

Як приклад технічної перевірки коректності аналітичного конвеєра у лістингу 3.3 наведено фрагмент SQL-запиту, який контролює, чи залишаються нормалізовані ознаки в допустимому інтервалі [0; 1].

Лістинг 3.3 – Фрагмент перевірки меж нормалізованих ознак

---

```
SELECT
  COUNT(*) FILTER (WHERE x1 < 0 OR x1 > 1) AS invalid_x1,
  COUNT(*) FILTER (WHERE x2 < 0 OR x2 > 1) AS invalid_x2,
  COUNT(*) FILTER (WHERE x3 < 0 OR x3 > 1) AS invalid_x3,
  COUNT(*) FILTER (WHERE x4 < 0 OR x4 > 1) AS invalid_x4,
  COUNT(*) FILTER (WHERE x5 < 0 OR x5 > 1) AS invalid_x5
FROM beneficiary_features_normalized;
```

---

Кінець лістингу 3.3

Окремо підготовлено файл `experiments.sql`, який містить запити для аналізу розподілу оцінок пріоритетності, кореляцій між нормалізованими ознаками та підсумковими балами, перевірки чутливості результатів до зміни ваг критеріїв і сценарного моделювання політик розподілу ресурсів. Детальний аналіз результатів цих запитів подано в підрозділі експериментальної перевірки.

Запуск системи організовано через файл `pipeline.sh`. Скрипт ініціалізує схему бази даних, завантажує тестові дані, створює аналітичні представлення та виконує перевірочні й експериментальні SQL-запити. У поєднанні з `Docker` це дозволяє відтворити структуру бази даних, аналітичні представлення, тестові дані та результати перевірки без ручного налаштування окремих компонентів.

Послідовність запуску SQL-скриптів у тестовому середовищі наведено в лістингу 3.4.

Отже, інтеграція, перевірка та запуск системи реалізовані як єдиний контрольований процес. База даних, `backend`-рівень, інтерфейс користувача,

перевірочні SQL-запити та контейнеризоване середовище працюють разом, забезпечуючи практичну працездатність системи й можливість повторного запуску.

Лістинг 3.4 – Фрагмент послідовного запуску аналітичного конвеєра

---

```

docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
init.sql
docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
operational_schema_layer.sql
docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
decision_schema_layer.sql
docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
scores.sql
docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
data_generator.sql
docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
decision_view.sql
docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
validation.sql
docker exec -i $DB_CONTAINER psql -U $DB_USER -d $DB_NAME <
experiments.sql

```

---

Кінець лістингу 3.4

### 3.6 Експериментальна перевірка та оцінка результатів

На експериментальному етапі перевіряється працездатність, коректність та інтерпретованість реалізованої інформаційно-аналітичної системи в синтетичних умовах, наближених до діяльності громадської організації. Основна мета перевірки полягає в тому, щоб встановити, чи коректно система формує оцінки пріоритетності бенефіціарів на основі MCDA-моделі, чи зберігає результати в заданому діапазоні та чи забезпечує змістовне ранжування неоднорідних груп бенефіціарів.

Формально експериментальна перевірка оцінює поведінку реалізованої функції оцінювання, яка перетворює нормалізований вектор ознак бенефіціара на інтегральну оцінку пріоритетності відповідно до формули (2.7). Для перевірки використано синтетичний набір даних, створений за допомогою файлу `data_generator.sql`. Це дозволяє перевірити SQL-перетворення,

нормалізацію, ранжування та аналіз чутливості без використання реальних персональних даних.

Кожен бенефіціар у тестовому наборі визначається базовими ознаками  $b_i=(h_i, y_i, d_i)$  (де  $h_i$  – величиною домогосподарства;  $y_i$  – дохід, а  $d_i$  – кількістю утриманців). Дохід формується за допомогою нелінійного стохастичного процесу:

$$y_i=200+\epsilon_i^2 \times 3200, \quad \epsilon_i \sim U(0,1).$$

Чисельність домогосподарства задається як дискретна величина  $h_i \sim U\{1,5\}$ , а кількість утриманців обмежується умовою  $d_i \leq h_{i-1}$ .

Це забезпечує логічну узгодженість тестових записів.

У таблиці 3.2 наведено основні характеристики експериментального набору даних та результати початкової технічної перевірки аналітичного конвеєра.

Таблиця 3.2 – Характеристика експериментального набору даних

Показник	Значення
Кількість бенефіціарів	10000
Кількість записів потреб	14035
Кількість записів вразливості	36796
Кількість записів наданої допомоги	16124
Некоректні нормалізовані значення ознак	0
Порожні значення інтегральної оцінки	0
Вироджені нормалізовані ознаки	Не виявлено

Наведені результати показують, що синтетичний набір містить достатню кількість записів для перевірки поведінки системи, а контрольні запити не виявили некоректних нормалізованих значень, порожніх інтегральних оцінок або вироджених ознак. Це узгоджується з обмеженнями коректності, визначеними у формулах (2.12), (2.13) та (2.14). Подальша перевірка зосереджується на розподілі підсумкових оцінок, коректності ранжування, впливі окремих ознак і чутливості системи до зміни ваг критеріїв.

На першому етапі оцінюється розподіл інтегральних балів пріоритетності, які формуються MCDA-моделлю. Для кожного бенефіціара підсумкова оцінка

обчислюється за формулою (2.7), а в базовому сценарії використовується рівномірний розподіл ваг, визначений у формулі (2.8).

Для оцінки поведінки результатів система обчислює мінімальне, максимальне та середнє значення підсумкового бала, а також стандартне відхилення:  $\min(S)$ ,  $\max(S)$ ,  $E[S]$  та  $\sigma_S$ .

У таблиці 3.3 наведено підсумкові статистичні характеристики інтегральних оцінок пріоритетності.

Таблиця 3.3 – Статистичні характеристики інтегральних оцінок пріоритетності

Показник	Значення
Кількість бенефіціарів	10000
Мінімальна оцінка	0,260
Максимальна оцінка	0,987
Середня оцінка	0,699
Стандартне відхилення	0,116

Дані таблиці 3.3 показують, що всі підсумкові оцінки залишаються у межах інтервалу  $[0, 1]$ , що відповідає обмеженню (2.13). Стандартне відхилення свідчить про наявність достатньої варіативності для практичного ранжування, а відмінність між мінімальним і максимальним значеннями показує, що виродження підсумкових оцінок не спостерігається.

Додатково розподіл балів поділяється на дискретні інтервали за допомогою рівномірної гистограми на проміжку  $[0,1]$  (3.1):

$$V_k = [(k-1)/10], [k/10], \quad (3.1)$$

де  $k \in \{1, \dots, 10\}$  – результати такого поділу наведено в таблиці 3.4.

Як видно з таблиці 3.4, найбільша частина бенефіціарів зосереджена в інтервалах від 0,6 до 0,8. Модель не зводить більшість оцінок до крайніх значень, а формує градацію пріоритетності, придатну для подальшого аналізу.

Невелика кількість записів з оцінками вище 0,9 утворює групу підвищеної пріоритетності.

Таблиця 3.4 – Розподіл інтегральних оцінок пріоритетності за інтервалами

Інтервал оцінки S	Кількість бенефіціарів	Частка, %
[0,2; 0,3]	2	0,02
[0,3; 0,4]	59	0,59
[0,4; 0,5]	429	4,29
[0,5; 0,6]	1587	15,87
[0,6; 0,7]	2777	27,77
[0,7; 0,8]	3005	30,05
[0,8; 0,9]	1870	18,70
[0,9; 1,0]	271	2,71

Для перевірки ранжування розглядаються бенефіціари з найбільшими та найменшими значеннями інтегральної оцінки  $S_i$ . Найвище значення становило 0,987, а найнижче – 0,260; верхні позиції рейтингу мають оцінки вище 0,955, тоді як нижні перебувають у діапазоні приблизно від 0,260 до 0,336. Це підтверджує здатність системи розрізняти бенефіціарів із різним рівнем пріоритетності.

Розподіл бенефіціарів за класами пріоритетності формується відповідно до правила класифікації, наведеного у формулі (2.9). Результати наведено в таблиці 3.5.

Таблиця 3.5 – Розподіл бенефіціарів за класами пріоритетності

Клас пріоритетності	Кількість бенефіціарів	Частка, %
Високий пріоритет	25	0,25
Середній пріоритет	7898	78,98
Низький пріоритет	2077	20,77

Дані таблиці 3.5 свідчать, що більшість бенефіціарів належить до середнього класу, тоді як високий пріоритет отримує обмежена група записів. Це є практично доцільним для системи підтримки прийняття рішень, оскільки високий клас може використовуватися для виділення найбільш критичних випадків.

Для оцінки пояснювальної валідності обчислюються коефіцієнти кореляції Пірсона (3.2) між нормалізованими ознаками та інтегральною оцінкою:

$$\rho(n_j, S) = \text{Cov}(n_j, S) / [(\sigma_{(n_j)} \cdot \sigma_S)], \quad (3.2)$$

де:  $n_j$  – нормалізована ознака;

$S$  – інтегральна оцінка пріоритетності;

$\text{Cov}(n_j, S)$  – коваріація між ознакою та оцінкою;

$\sigma_{(n_j)}$  і  $\sigma_S$  – стандартні відхилення відповідних величин.

Формула (3.2) використовується для перевірки узгодженості напряму впливу ознак із логікою моделі. Оскільки  $n_1$  та  $n_5$  є вже перетвореними показниками низької економічної спроможності й низького рівня попередньо отриманої допомоги, для всіх п'яти ознак очікується додатний зв'язок із підсумковою оцінкою.

Результати кореляційного аналізу наведено в таблиці 3.6.

Таблиця 3.6 – Кореляція нормалізованих ознак з інтегральною оцінкою

Ознака	Змістова інтерпретація	Кореляція з S
$n_1$	Низька економічна спроможність	0,176
$n_2$	Демографічне навантаження	0,518
$n_3$	Рівень вразливості	0,595
$n_4$	Інтенсивність потреби	0,535
$n_5$	Низький рівень попередньо отриманої допомоги	0,184

Як видно з таблиці 3.6, усі нормалізовані ознаки мають додатний зв'язок із підсумковою оцінкою. Найбільшу кореляцію мають рівень вразливості, інтенсивність потреби та демографічне навантаження, що узгоджується зі змістовною логікою системи. Ознаки  $n_1$  та  $n_5$  мають слабший, але додатний вплив, що відповідає їх ролі як допоміжних факторів.

Наступний етап – це оцінка реакції системи на зміну ваг критеріїв. У базовому сценарії використовується рівномірний розподіл ваг визначений (2.7).

Для перевірки чутливості задається альтернативний вектор ваг:

$$w'=(0,10; 0,15; 0,20; 0,40; 0,15),$$

у якому більший вплив на підсумковий результат має показник інтенсивності потреби  $n_4$ .

Зміна підсумкової оцінки для кожного бенефіціара визначається з (3.3):

$$\Delta S_i=S_i(w')-S_i(w). \quad (3.3)$$

Після цього перевіряється зміна рангової позиції (3.4):

$$\Delta r_i=|r_i(w)-r_i(w')|. \quad (3.4)$$

Для оцінки загальної узгодженості між базовим та альтернативним рейтингами використовується кореляція між векторами рангів: (3.5)

$$\rho_r=\text{corr}[r(w), r(w')]. \quad (3.5)$$

Формули (3.3)-(3.5) дозволяють оцінити як зміну числових балів, так і стабільність загальної структури ранжування.

Узагальнені результати аналізу чутливості наведено в таблиці 3.7.

Таблиця 3.7 – Результати аналізу чутливості до зміни ваг критеріїв

Показник	Значення
Кореляція між базовим та альтернативним ранжуванням	0,942
Найбільша зафіксована зміна рангу	5068
Найбільше зафіксоване зниження оцінки	-0,198
Кількість проаналізованих найбільш чутливих записів	20

Дані таблиці 3.7 показують, що зміна ваг впливає на окремі оцінки та позиції бенефіціарів. Це підтверджується зміною балів за формулою (3.3) та зміною рангів за формулою (3.4). Водночас кореляція між базовим та альтернативним ранжуванням, обчислена за формулою (3.5), дорівнює 0,942, що свідчить про збереження загальної структури рейтингу. Отже, система

чутлива до зміни політики оцінювання на рівні окремих записів, але сформоване ранжування загалом залишається узгодженим.

Додатково перевірено, як сформовані оцінки можуть використовуватися у прикладних сценаріях діяльності громадської організації. Ця перевірка не є окремою моделлю розподілу допомоги, а використовується для оцінки практичної придатності рейтингу та змістовної узгодженості результатів.

Перший сценарій пов'язаний із вибором обмеженої кількості бенефіціарів із найвищими значеннями пріоритетності. Стратегія модельованого розподілу визначається як вибір множини бенефіціарів з найвищими значеннями інтегральної оцінки (3.6):

$$A_K = \{b_i \mid S_i \in T_{op}K(S, K)\}, \quad (3.6)$$

де:  $A_K$  – множина відібраних бенефіціарів;

$S_i$  – інтегральна оцінка пріоритетності бенефіціара;

$K=100$  – кількість записів у пріоритетній групі.

Формула (3.6) використовується для перевірки детермінованого формування пріоритетної групи за однакових вхідних даних і параметрів оцінювання.

Другий сценарій спрямований на перевірку узгодженості оцінок із соціально-економічними характеристиками. Для цього бенефіціари групуються за рівнем доходу відповідно до категорій:

$$G(b_i) \in \{\text{«низький»}, \text{«середній»}, \text{«високий»}\},$$

які сформовано у SQL-запиті експериментальної перевірки. Для кожної групи обчислюється середнє значення пріоритетності:

$$\mu_g = E [S \mid G = g].$$

Результати групування наведено в таблиці 3.8.

Як видно з таблиці 3.8, група з високим доходом має нижчу середню оцінку пріоритетності порівняно з групами низького та середнього доходу. Це

підтверджує, що економічна характеристика враховується системою в очікуваному напрямі. Водночас близькість середніх значень для груп низького та середнього доходу показує, що підсумкова оцінка формується не лише доходом, а одночасним впливом кількох критеріїв.

Таблиця 3.8 – Середні оцінки пріоритетності за групами доходу

Група доходу	Кількість бенефіціарів	Середня оцінка
Низький дохід	1331	0,708
Середній дохід	5714	0,710
Високий дохід	2955	0,674

Отже, додаткова сценарна перевірка показує, що результати роботи системи можуть використовуватися для індивідуального ранжування, формування пріоритетної групи отримувачів допомоги та контролю змістовної узгодженості оцінок із базовими характеристиками бенефіціарів.

## ЗАГАЛЬНІ ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

У кваліфікаційній роботі виконано проектування та реалізацію інформаційної системи збору й аналізу даних для підтримки діяльності громадської організації. Розроблена система поєднує реляційну базу даних, SQL-орієнтовану аналітичну обробку, модель багатокритеріального оцінювання, серверну частину та користувацький інтерфейс. У результаті сформовано прикладне рішення, яке дає змогу впорядкувати дані про бенефіціарів, виконувати їх аналітичне оцінювання, визначати рівень пріоритетності та подавати результати у зручній для користувача формі.

За результатами виконання роботи сформульовано наступні висновки:

– сформовано реляційну модель даних для зберігання інформації про бенефіціарів, потреби, вразливості та надану допомогу. У структурі бази даних виділено основні сутності предметної області, що забезпечують цілісне зберігання інформації, зменшують дублювання даних і створюють основу для подальшої аналітичної обробки;

– розроблено SQL-орієнтований аналітичний конвеєр формування похідних ознак на основі операційних даних. Він забезпечує перехід від первинних записів у базі даних до аналітичних показників, які можуть бути використані для оцінювання стану бенефіціарів і підтримки прийняття рішень.

– забезпечено нормалізацію різнорідних показників для їх коректного порівняння у межах моделі оцінювання. У роботі враховано критерії прямої та оберненої дії, що дало змогу привести різні за змістом і масштабом показники до єдиної форми для подальшого багатокритеріального аналізу;

– побудовано модель багатокритеріального оцінювання пріоритетності бенефіціарів. На основі нормалізованих критеріїв і вагових коефіцієнтів система формує інтегральну оцінку, виконує ранжування отримувачів допомоги та визначає рівень їхньої пріоритетності;

– реалізовано механізм пояснюваності результатів через визначення внеску окремих критеріїв у підсумкову оцінку. Це дає змогу не лише отримати

числовий результат ранжування, а й зрозуміти, які саме фактори найбільше вплинули на підсумкову пріоритетність конкретного бенефіціара;

– реалізовано серверну частину та користувацький інтерфейс системи. Інтерфейс дає змогу вводити дані, переглядати аналітичну панель, таблицю рішень, категорії пріоритетності, пояснення результатів і профіль окремого бенефіціара. Серверна частина забезпечує обробку запитів, взаємодію з базою даних і передачу результатів аналітичної обробки до користувацького інтерфейсу;

– проведено експериментальну перевірку коректності, стабільності, чутливості та відтворюваності роботи системи. Перевірено коректність нормалізованих значень, розподіл інтегральних оцінок, узгодженість ранжування, вплив зміни ваг критеріїв і повторюваність отриманих результатів. Отримані результати підтвердили працездатність системи та коректність основних аналітичних перетворень.

Практичне значення роботи полягає в тому, що запропонована система може бути використана як основа для впорядкування процесів збору, зберігання та аналізу даних у громадській організації. Вона не замінює відповідальних працівників, але надає їм додаткову аналітичну основу для прийняття більш послідовних і обґрунтованих рішень щодо розподілу допомоги.

Разом з тим робота має певні обмеження. Перевірка системи виконувалася на підготовленому наборі даних, тому перед практичним використанням її необхідно адаптувати до реальних процесів конкретної організації. Зокрема, потрібно уточнити критерії оцінювання, погодити вагові коефіцієнти з фахівцями, визначити правила оновлення даних і забезпечити належний контроль доступу до персональної інформації.

Подальший розвиток системи може передбачати підключення реальних джерел даних, розширення набору критеріїв оцінювання, удосконалення інтерфейсу користувача, додавання механізмів аудиту рішень та інтеграцію із зовнішніми інформаційними системами громадської організації.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Василенко Д. В. Підтримка прийняття рішень на основі інформаційно-аналітичного забезпечення у системі публічного управління. URL: <https://aprus.khpi.edu.ua/article/view/336934> (дата звернення: 10.01.2026).

2. Салата Н. Дослідження питань цифровізації соціального обслуговування в Україні. URL: <https://srso.udpu.edu.ua/article/view/294343> (дата звернення: 10.01.2026).

3. Ihnatenko, K., Sadzaglishvili, Sh. The digitalization of social services in response to the war in Ukraine. URL: <https://surl.li/kchbhh> (access date: 10.01.2026).

4. Скочко М. Цифрові та інформаційні технології у системі соціального захисту. Соціальна робота та соціальна освіта. 2026. URL: <https://srso.udpu.edu.ua/article/view/355754> (дата звернення: 10.01.2026).

5. Бражко О., Січкач Н. Модернізація організаційного механізму державного управління системою соціальних послуг у контексті цифровізації. URL: <https://litopys.volyn.ua/index.php/litopys/article/view/569> (дата звернення: 10.01.2026).

6. Каплун Є., Толстоносів Д., Дерев'яно М. Розроблення математичної моделі інформаційно-аналітичної системи для підтримки та прийняття рішень щодо логістичного забезпечення Національної гвардії України на основі онтологічного підходу. URL: <https://znp.nangu.edu.ua/article/view/307883> (дата звернення: 10.01.2026).

7. Данилюк М., Дмитришин М., Горан Т. Цифровізація соціальних послуг. URL: <https://doi.org/10.37634/efp.2023.8.9> (дата звернення: 20.01.2026).

8. Skochko, M., Salata, N. Impact of Digital Technologies on the Accessibility of Social Services for Vulnerable Population Categories. URL: <https://journals.uran.ua/swe/article/view/349235> (access date: 20.01.2026).

9. Хлівнюк Т. П., Кіріяк А. І. Соціальна інклюзія в контексті цифровізації: можливості та ризики для соціальної держави. URL: <https://surl.li/ecbner> (дата звернення: 20.01.2026).

10. Архипова С., Белова І., Тарануха Т. Цифрові інструменти персоналізації соціальних послуг для внутрішньо переміщених осіб в Україні. URL: <https://ven.chdtu.edu.ua/article/view/353841> (дата звернення: 20.01.2026).

11. Мулеса О. Ю., Горват П. П., Єгорченков О. В., Імре Ю.Ю., Ференс Д. Я., Коціпак В. О. Людино-машинні системи підтримки прийняття рішень з числовими каналами. URL: <https://surl.li/hpsrvu> (дата звернення: 20.01.2026).

12. Бідюк П. І., Тимощук О. Л., Коваленко А. Є., Коршевніук Л. О. Системи і методи підтримки прийняття рішень. URL: <https://ela.kpi.ua/items/d242b18b-3d30-4754-b774-ec79c1fdeea8> (дата звернення: 28.01.2026).

13. PostgreSQL Global Development Group. PostgreSQL Documentation. 2025. URL: <https://www.postgresql.org/docs/> (access date: 12.02.2026).

14. Василенко Д. В. Система підтримки прийняття рішень у публічному управлінні на основі інформаційного та аналітичного забезпечення. URL: <https://chasopys-ppp.dp.ua/index.php/chasopys/article/view/562> (дата звернення: 12.02.2026).

15. Ткаченко О. А., Ткаченко О. І., Делант О. О. Система підтримки прийняття рішень щодо вибору велосипеда. URL: <https://its.istu.edu.ua/index.php/ITS/article/view/25> (дата звернення: 22.03.2026).

16. Коваленко О., Додон О. Інформаційні технології в системі соціального захисту. URL: <https://surl.li/muaizs> (дата звернення: 22.03.2026).

17. Авраменко Д. Ф. Функціональна модель системи підтримки прийняття рішень в системі стратегічних комунікацій Збройних Сил України. URL: <https://sit.nuou.org.ua/article/view/291220> (дата звернення: 22.03.2026).

18. European Union. General Data Protection Regulation (GDPR). URL: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> (access date: 05.04.2026).

19. Bootstrap Team. Bootstrap Documentation. URL: <https://surl.lu/ztuudg> (access date: 05.04.2026).

20. Гриценко Г. М. Цифровізація соціальних послуг: проблеми можливостей та компетенції. URL: <https://doi.org/10.33216/2218-5461/2024-47-1-36-48> (дата звернення: 20.04.2026).

21. Донець В., Шматков С. Методи аналізу інформативності в медичних системах підтримки прийняття рішень. URL: <https://surl.li/cqaozr> (дата звернення: 20.04.2026).

22. Козуб В. Інформаційна підтримка прийняття рішень із застосуванням технологій розподіленого штучного інтелекту. URL: <https://surl.li/gzgpkk> (дата звернення: 20.04.2026).

23. Бойко А. О., Шулімова Д. Д. Модель загроз систем підтримки прийняття рішень, що функціонують на основі елементів штучного інтелекту. URL: <https://tit.dut.edu.ua/index.php/telecommunication/article/view/2476> (дата звернення: 20.04.2026).

24. Docker Inc. Docker Documentation. URL: <https://docs.docker.com/> (access date: 20.04.2026).

25. Терлецький Т. В., Кайдик О. Л. Кваліфікаційна робота: методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Інформаційні системи та технології охорони і безпеки» галузі знань 12 Інформаційні технології спеціальності 126 Інформаційні системи та технології денної та заочної форм навчання. Луцьк: ЛНТУ, 2025. 53 с.