Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

«Розробка та дослідження методів пріоритезації виплат системи обліку пошкодженого житла»
(тема)

Виконав:

здобувач 2 року навчання,
групи ІТПм-24-1

Вадим РЕБРОВ
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Інформаційні технології проєктування
(повна назва освітньої програми)

Керівник доц. каф. СТ Тетяна БІЛОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____ Ігор ГРЕБЕННІК
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Інформаційні технології проєктування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. Кафедри_____
(підпис)

«___» _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві РЕБРОВУ Вадиму Станіславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка та дослідження методів пріоритезації виплат системи обліку пошкодженого житла»

затверджена наказом університету від 24 листопада 2025р. № 1058 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 14 грудня 2025 р.

3. Вихідні дані до роботи розробка інформаційної технології пріоритезації виплат та реалізація її у вигляді компоненту системи обліку пошкодженого житла

4. Перелік питань, що потрібно опрацювати в роботі Аналіз предметної області. Аналіз предметної області, яка визначає діяльність підприємства. Методи обчислення пріоритету виплат. Лінійна модель. Алгоритми пріоритетів. Методи «снігової кулі» та «лавини». Метод аналізу ієрархій. Метод нечіткої логіки. Метод машинного навчання. Теорія прийняття рішень. Постановка задачі. Дослідження існуючих підходів, методів, засобів для вирішення задач пріоритезації. Огляд підходів до вирішення задачі пріоритезації в науковій літературі. Доцільність застосування нечіткої логіки для задач пріоритезації в умовах невизначеності. Опис системи нечіткого виведення. Порівняння систем мамдані та сугено. Розробка інформаційної технології вирішення задачі. Розробка функціональних вимог до

технології пріоритезації. Розробка моделі потоків даних технології пріоритезації. Розробка вимог до інтерфейсу клієнтської частини технології пріоритезації. Опис архітектури розробленої системи. Фізичне моделювання даних системи. Імплементація нечіткої логіки у технологію пріоритезації. Тестування технології пріоритезації. Аналіз обмежень поточної реалізації. Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Функція приналежності desctruction. Функція приналежності residents. Функція приналежності age. Функція приналежності disability_group. Функція приналежності is_large_family. Функція приналежності priority.Контекстна діаграма IDEF0. Діаграма декомпозиції IDEF0. Контекста діаграма DFD. Діаграма декомпозиції DFD. Use-case діаграма IT пріоритезації виплат. Діаграма класів IT пріоритезації виплат. Діаграма послідовностей для створення пріоритетної виплати. Архітектура компоненту пріоритезації системи обліку пошкодженого житла. Структура колекції виплат у БД. Дані про власника та житло. Логи сервісу. Список виплат. Конфігурація навантаження. Графік навантажень.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основний розділ	доц. каф. СТ Білова Тетяна Георгіївна		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Аналіз предметної області.	01.11.2025	Виконано
3	Дослідження існуючих підходів, методів, засобів для вирішення задач пріоритезації	20.11.2025	Виконано
4	Розробка інформаційної технології вирішення задачі	25..11.2025	Виконано
5	Тестування технології	30.11.2025	Виконано
6	Представлення на рецезування	13.12.2025	Виконано
7	Представлення кваліфікаційої роботи	14.12.2025	Виконано

Дата видачі завдання 24.11.2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. каф. СТ Тетяна БІЛОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 76 с., 3 табл., 20 рис., 1 додаток.

АВТОМАТИЗАЦІЯ ВИПЛАТ, АНАЛІЗ ДАНИХ, ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, МОДЕЛЬ МАМДАНІ, НЕЧІТКА ЛОГІКА, ПРІОРИТИЗАЦІЯ ВИПЛАТ, СИСТЕМА ОБЛІКУ ПОШКОДЖЕНОГО ЖИТЛА.

Сучасні інформаційні системи соціального спрямування потребують автоматизації процесів прийняття рішень, пов’язаних із визначенням черговості виплат компенсацій. У разі великої кількості заявок та обмежених фінансових ресурсів виникає складність у встановленні справедливого порядку виплат, що зумовлює необхідність застосування методів нечіткої логіки.

Об’єктом дослідження є процес пріоритезації виплат у системі обліку пошкодженого житла. Предметом дослідження є методи нечіткого логічного виведення та формалізації соціальних і технічних критеріїв для визначення пріоритету виплат компенсацій.

Метою роботи є розробка інформаційної технології пріоритезації виплат та реалізація її у вигляді компоненту системи обліку пошкодженого житла.

Методами дослідження є аналіз предметної області, методи нечіткої логіки типу Мамдані та Сугено, математичне моделювання, нормування показників, фазифікація та дефаззифікація даних. Для реалізації програмної частини використано засоби Java Spring Framework, Thymeleaf та СУБД MongoDB.

Наукова новизна роботи полягає у створенні інформаційної технолоії автоматичного визначення пріоритетів виплат на основі нечіткого моделювання соціально-економічних параметрів, що забезпечує об’єктивність, прозорість та гнучкість прийняття рішень. Розроблений підхід дозволяє скоротити вплив людського фактора, підвищити швидкість і точність формування списків компенсацій, а також може бути використаний у благодійних організаціях та державних програмах соціального відновлення.

ABSTRACT

Explanatory note of the qualification work:: 76 pages, 3 tables, 20 pic., 1 appendix.

DAMAGED DWELLING ACCOUNTING SYSTEM, DATA ANALYSIS, FUZZY LOGIC, INFORMATION TECHNOLOGY, MAMDANI MODEL, PAYMENT AUTOMATION, PAYMENT PRIORITIZATION.

Modern socially-oriented information systems require automation of decision-making processes related to determining the priority of compensation payments. Given the large number of applications and limited financial resources, establishing a fair payment order becomes difficult, necessitating the application of fuzzy logic methods.

The object of the study is the process of payment prioritization within the damaged housing accounting system.

The subject of the study is the methods of fuzzy inference and the formalization of social and technical criteria for determining the priority of compensation payments.

The purpose of the work is to develop an information technology for payment prioritization and implement it as a component of the damaged housing accounting system.

The research methods include domain analysis, Mamdani and Sugeno fuzzy logic methods, mathematical modeling, indicator normalization, as well as data fuzzification and defuzzification. The software component was implemented using the Java Spring Framework, Thymeleaf, and MongoDB DBMS.

The scientific novelty of the work lies in the creation of an information technology for the automatic determination of payment priorities based on fuzzy modeling of socio-economic parameters. This ensures objectivity, transparency, and flexibility in decision-making. The developed approach allows for reducing the influence of the human factor, increasing the speed and accuracy of generating compensation lists, and can be utilized by charitable organizations and state social recovery programs.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – База даних

ІС – Інформаційна система

ІТ – Інформаційна технологія

ПЗ – Програмне забезпечення

СУБД – Система управління базами даних

ТПР – Теорія прийняття рішень

ФН – Функція належності

API – Application Programming Interface

DFD – Data Flow Diagram

FCL – Fuzzy Control Language

HTTP – HyperText Transfer Protocol

IBAN – International Bank Account Number

IDEF0 – Integration Definition for Function Modeling

JSON – JavaScript Object Notation

ML – Machine Learning

PDF – Portable Document Format

REST – Representational State Transfer

UML – Unified Modeling Language

ВСТУП

Сучасні соціально-економічні процеси в Україні характеризуються високим рівнем невизначеності, спричиненої як військовими діями, так і масштабними руйнуваннями житлової інфраструктури. Значна кількість домогосподарств зазнає пошкоджень або повного знищення житла, що потребує оперативної організації процесів обліку, оцінювання та компенсації завданих збитків. У таких умовах особливо важливим є створення ефективних інформаційних систем, здатних забезпечити прозорий і справедливий механізм розподілу фінансової допомоги постраждалим громадянам.

Однією з ключових проблем функціонування систем компенсацій є визначення пріоритетності виплат, коли кількість заявок значно перевищує доступний бюджет, а соціально вразливі групи населення потребують першочергової підтримки. У традиційних підходах процес прийняття рішень покладається на бухгалтерів або відповідальних осіб, що створює ризики суб’єктивності, перенавантаження та ймовірності помилок. Крім того, значна частина критеріїв, що впливають на порядок мають нечіткий або якісний характер, що ускладнює їх формалізацію класичними математичними методами. Тому виникає потреба у впровадженні інтелектуальних методів підтримки прийняття рішень, здатних враховувати якісні фактори та працювати з нечіткими межами показників.

Задача під час проведення кваліфікаційної роботи – розробити інформаційну технологію пріоритезації виплат та реалізувати її у вигляді компоненту системи обліку пошкодженого житла.

Сфера застосування – благодійні фонди та некомерційні організації, що спеціалізуються на гуманітарній та фінансовій допомозі. Результати кваліфікаційної роботи апробовано на IX Всеукраїнській студентській науковій конференції «Розвиток сучасної науки: актуальні питання теорії та практики» [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області оцінки пошкодженого житла

Однією з найбільш складних задач у сфері бухгалтерського обліку та фінансового адміністрування є визначення пріоритетності виплат, особливо в питаннях гуманітарного характеру: коли кількість заявок на фінансову допомогу або компенсацію є великою, а ресурс обмежений, постає потреба встановити справедливу черговість виплат. У таких випадках суб'єктивне рішення бухгалтера може призвести до неточностей, затримок або конфліктів, особливо якщо порядок виплат має критичне значення як, наприклад, у програмах соціальної допомоги, страхових компенсаціях чи відшкодуванні збитків постраждалим власникам житла.

Ця проблема особливо актуальна для України, де через бойові дії значна кількість житлових об'єктів була пошкоджена або зруйнована. Держава реалізує програму «єВідновлення» у застосунку «Дія», що дозволяє подати заявку на компенсацію [2], однак механізм визначення пріоритету між заявниками відсутній. Через це розподіл коштів між великою кількістю заявок залишається тривалим і часто залежить від людського фактору.

Щоб розв'язати цю проблему, потрібно створити інформаційну технологію (ІТ) пріоритезації виплат, який потрібно інтегрувати в існуючу інформаційну систему обліку пошкодженого житла у вигляді компоненту. Це дає змогу забезпечити прозорість і об'єктивність при розподілі благодійних внесків.

Інформаційна система обліку пошкодженого житла була розроблена з метою автоматизації процесів реєстрації заявок на відновлення житла, створення комісій з оцінки пошкоджень, планування оглядів та фіксації результатів перевірки. Система використовується у благодійних та

гуманітарних організаціях, що здійснюють допомогу постраждалим власникам житла, і є частиною ширшої екосистеми обліку компенсацій.

Базова система забезпечує:

– реєстрацію заявок власників житла через веб–інтерфейс;

– ведення обліку комісій за населеними пунктами;

– збереження чек–листів пошкоджень після огляду;

– формування загального списку підтверджених заявок зі статусом

Однак на поточному етапі система не містить механізму автоматизованого розподілу коштів. Після проведення оцінки житла всі підтверджені заявки надходять до бухгалтерії, де фахівці вручну формують списки для виплат. Такий підхід має низку недоліків:

– значні часові витрати на опрацювання великої кількості заявок;

– високий ризик суб’єктивності у визначенні черговості компенсацій;

– складність у врахуванні соціальних критеріїв (вік, інвалідність, кількість мешканців);

– відсутність прозорої логіки визначення пріоритетів;

– труднощі з формуванням звітності та поясненням причин вибору черговості виплат.

Відсутність чіткої формалізованої моделі визначення пріоритетів призводить до затримок у виплатах і знижує довіру з боку меценатів і постраждалих осіб.

Для визначення ступеня пріоритетності доцільно враховувати такі показники:

– вік власника (особи похилого віку мають підвищену соціальну вразливість, тому в алгоритмі їхній коефіцієнт має більшу вагу)

– кількість зареєстрованих осіб (чим більше людей проживає за адресою, тим вищий пріоритет, адже відновлення такого житла впливає на більшу кількість осіб);

– наявність групи інвалідності (передбачено три групи: I група – найвищий коефіцієнт пріоритету; II група – середній коефіцієнт; III група – нижчий, але вищий за відсутність інвалідності)

– ознака багатодітності (так – більше двох дітей включно, ні – немає дітей або одна дитина у сім'ї);

– загальний ступінь пошкодження житла (визначається автоматично на основі чек-ліста пошкоджень, який заповнює комісія після огляду. Для кожного елемента чек–листа задається бал, наприклад, вікно – 2 бали, дах – 5 балів, несуча стіна – 10 балів. Загальна сума балів нормується до шкали від від 0 до 10).

1.2 Методи обчислення пріоритету виплат

Проблема визначення пріоритетності виплат у системах соціального або гуманітарного спрямування є задачею багатокритеріального оцінювання. Її суть полягає у тому, щоб для кожного об'єкта (у даному випадку пошкодженого житла) обчислити інтегральний показник, який відображає ступінь соціальної та матеріальної потреби у відновленні. Для розв'язання таких задач існує кілька підходів, що відрізняються складністю реалізації, точністю результатів та вимогами до вихідних даних.

1.2.1 Лінійна модель

Найпростішим способом є лінійна модель з ваговими коефіцієнтами, у якій кожному параметру (вік власника, ступінь пошкодження, кількість мешканців тощо) призначається власна вага залежно від його важливості. У цій моделі пріоритет розраховується як зважена сума нормованих параметрів. Такий підхід дозволяє швидко реалізувати алгоритм, має високу прозорість для користувачів і легко піддається налаштуванню.

В даному випадку пріоритет можна обчислити як зважену суму нормованих параметрів. Для кожного критерію x_i задається вага w_i, яка відображає його важливість. Інтегральний показник визначається формулою:

$P = \sum_{i=1}^{n} w_i \cdot x_i$, де $\sum_{i=1}^{n} w_i = 1$, а всі x_i попередньо нормуються до інтервалу [0, 1].

Такий підхід простий у реалізації, але не враховує взаємозалежність факторів і не дозволяє описати нечіткі або якісні характеристики, наприклад «похилого віку» чи «значного пошкодження».

1.2.2 Алгоритми пріоритетів. Методи «снігової кулі» та «лавини»

Алгоритми пріоритетів застосовуються у тих випадках, коли необхідно визначити черговість виконання фінансових зобов'язань, враховуючи декілька факторів або критеріїв, що характеризують їх важливість чи вплив на загальну фінансову стійкість системи. Найбільш відомими підходами є метод «снігової кулі» (Debt Snowball) та метод «лавини» (Debt Avalanche) [3], які спочатку були розроблені у сфері управління заборгованістю, але можуть бути адаптовані до процесів пріоритизації виплат. Метод «снігової кулі» передбачає впорядкування платежів за величиною їхнього розміру: першочергово обслуговуються найменші зобов'язання, що дозволяє швидко зменшувати кількість активних платежів і підвищує мотивацію завдяки швидкому досягненню проміжних результатів. Метод «лавини» орієнтований на мінімізацію загальних витрат, тому виплати здійснюються у напрямку зобов'язань із найвищим значенням відсоткової ставки або фінансових втрат, що дозволяє зменшити загальний розмір витрат у довгостроковій перспективі. У контексті пріоритизації компенсацій такі підходи можуть бути модифіковані з урахуванням специфічних параметрів соціальної вразливості або критичності стану житла. Алгоритм «снігової кулі» може бути інтерпретований як присвоєння вищого пріоритету заявкам з найменшими сумами компенсацій, тоді як «лавина» – як орієнтація на найбільш критичні

показники ризику або збитків. Завдяки своїй простоті та інтуїтивності ці методи можуть застосовуватися як частина експертної системи пріоритизації або як базові евристики у випадках, де немає можливості побудови складної математичної моделі.

1.2.3 Метод аналізу ієрархій

Наступним підходом є метод аналізу ієрархій (AHP – Analytic Hierarchy Process). Його суть полягає у побудові ієрархічної структури критеріїв та попарному порівнянні їх за важливістю [4]. У результаті формується матриця відносних пріоритетів, на основі якої обчислюється інтегральний бал для кожної заявки.

Для кожної пари критеріїв *(i, j)* формується матриця порівнянь $A = [a_{ij}]$, де a_{ij} відображає відносну важливість критерію i над j. Елементи матриці задовольняють умову:

$$a_{ij} = \frac{1}{a_{ji}}, a_{ii} = 1.$$

Для наочності наведено приклад застосування методу для визначення вагових коефіцієнтів трьох критеріїв пріоритезації виплат:

– K1 (ступінь руйнування житла);

– K2 (група інвалідності);

– K3 (кількість мешканців).

Експерт виконує попарне порівняння, використовуючи шкалу Сааті (від 1 до 9). Припустимо, що в умовах обмеженого бюджету:

– K1 є дещо важливішим за K2, (оцінка 3) і значно важливішим за K3 (оцінка 5), оскільки фізичний стан житла є першочерговим фактором;

– K2 є більш пріоритетною за K3 (оцінка 2).

Матриця порівнянь A матиме такий вигляд:

$$A = \begin{pmatrix} 1 & 3 & 5 \\ \frac{1}{3} & 1 & 2 \\ \frac{1}{5} & \frac{1}{2} & 1 \end{pmatrix}.$$

Власний вектор *w*, який відповідає найбільшому власному значенню λ_{max} матриці A, нормується і визначає ваги критеріїв $Aw = \lambda_{max} w$. Після проведення обчислень отримуємо вектор пріоритетів:

$$w = (0.65; 0.23; 0.12).$$

Це означає, що в даній моделі критерій «Ступінь пошкодження» отримав вагу 65%, «Група інвалідності» – 23%, а «Кількість мешканців» – 12%.

Цей метод забезпечує більшу обґрунтованість вибору ваг, оскільки базується на експертних оцінках у парних порівняннях, а також дозволяє виявляти непослідовність у судженнях. Його використання доцільне у випадках, коли існує достатньо експертів, здатних оцінити вплив кожного параметра на загальний результат. Недоліком методу є складність для реалізації в автоматизованій системі з великим числом критеріїв.

1.2.4 Метод нечіткої логіки

Більш гнучким і сучасним підходом є застосування нечіткої логіки (fuzzy logic), яка дозволяє враховувати невизначеність і лінгвістичні оцінки параметрів [5]. Наприклад, ступінь пошкодження може бути описаний не лише числом, а й термінами «незначний», «середній», «значний», кожен з яких має власну функцію належності. Аналогічно, для віку власника або кількості мешканців можуть бути задані нечіткі правила типу: «якщо вік високий і

пошкодження високі, то пріоритет великий». Система нечіткого висновку дозволяє моделювати реальні соціальні критерії у природній для експертів формі.

У нечітких системах Мамдані та Сугено основою опису лінгвістичних змінних є функції належності, які задають ступінь приналежності значення до певного терму (наприклад, «низький», «середній», «високий»). Найпоширеними є трикутні та трапецевидні функції належності завдяки простоті математичного опису та інтерпретації.

Трикутна функція належності визначається трьома параметрами (a, b, c).

Математично вона описується формулою:

$$\mu A(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x \le b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \ge c \end{cases},$$

де a – початок зростання функції;

b – її максимальне значення;

c – кінець спадання.

Ця функція використовується у випадках, коли значення змінної має чіткий центр (найтиповіше значення), а належність зменшується симетрично або майже симетрично в обидва боки. Трапецевидна функція належності задається чотирма параметрами (a, b, c, d).

Її форма передбачає наявність плато, де ступінь належності дорівнює 1.

Математичний вираз:

$$\mu A(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x \le b \\ 1, & b < x \le c \\ \frac{d-x}{d-c}, & c < x \le d \\ 0, & x > d \end{cases},$$

де a – початок зростання;

b – початок плато;

c – кінець плато;

d – кінець спадання функції.

Для отримання числового значення вихідної змінної використовується метод центра ваги.

Таким чином, нечітка модель дозволяє відобразити не тільки кількісні, а й якісні характеристики, наприклад ступінь пошкодження («низький», «середній», «високий»), вік (молодий, середній, похилий). Результат обчислення пріоритету є плавним, а не дискретним, що дозволяє уникнути різких переходів між рівнями важливості.

1.2.5 Метод машинного навчання

Ще одним підходом, який можна розглядати для подальшого розвитку системи, є методи машинного навчання, які дозволяють автоматично виявляти залежності між параметрами та пріоритетом на основі історичних даних про попередні виплати.

Найчастіше в таких системах застосовуються ансамблеві алгоритми, зокрема XGBoost [6] та LightGBM [7], які здатні враховувати нелінійні зв’язки між вхідними змінними та забезпечують високу точність прогнозування. Такі моделі можуть оцінювати ймовірність потреби у терміновій виплаті, визначати ступінь пріоритету для різних категорій заявників або прогнозувати ризики затримки у процесі розподілу коштів.

У загальному випадку прогноз пріоритету можна описати як функцію:

$$P = f\left(x_{1,}x_{2,} \dots x_n; \theta\right),$$

де θ – набір параметрів, які оптимізуються під час навчання моделі.

Крім того, деревоподібні алгоритми, наприклад Decision Trees і Random Forest, дозволяють інтерпретувати вплив кожного фактора на рішення [8], що є важливою перевагою в соціально чутливих системах. У складніших випадках використовуються нейронні мережі, які дають змогу моделювати багатовимірні залежності та працювати з великими масивами даних, але потребують більшої кількості навчальної інформації та менш прозорі для пояснення.

1.2.6 Теорія прийняття рішень

Теорія прийняття рішень розглядає процес вибору оптимальної альтернативи з множини можливих варіантів за умов невизначеності або багатокритеріальності, що робить її важливим інструментом у задачах пріоритезації виплат [9].

У системах фінансового розподілу коштів вона дає можливість формалізувати оцінювання кожної виплати, враховуючи як кількісні, так і якісні показники – терміновість, соціальну значущість, ступінь пошкодження житла, очікувані наслідки затримки та обмеженість ресурсів. Основою цього підходу є побудова матриці рішень, де кожна альтернатива характеризується набором критеріїв, а значення критеріїв можуть бути як точними, так і нечіткими. На практиці рішення може обиратися за допомогою класичних критеріїв теорії рішень, таких як критерій максиміну, критерій мінімаксу, критерій Гурвіца або критерій Лапласа, які дозволяють враховувати різний рівень ризиковості та обачності суб’єкта, що приймає рішення.

У випадку пріоритезації виплат ці критерії можуть визначати, чи повинні першочергово фінансуватися найбільш критичні випадки, чи ті, що мають максимальну соціальну корисність, чи ті, де затримка створює найбільші ризики. Теорія прийняття рішень також охоплює методи корисності, що дозволяють оцінювати альтернативи через функції корисності, де кожному значенню критерію приписується ступінь корисності, який

відображає його вплив на загальний результат. Завдяки цьому підходу система може формувати пріоритети з урахуванням індивідуальних або нормативних вимог, забезпечуючи логічну прозорість процесу та можливість обґрунтування прийнятих рішень.

1.3 Постановка задачі

Метою роботи є розробка інформаційної технології пріоритезації виплат та реалізація її у вигляді компоненту системи обліку пошкодженого житла.

В рамках поставленої мети слід вирішити наступні задачі:

– провести дослідження шляхом аналізу наявних підходів в наукових джерелах для вирішення задачі знаходження пріоритету;

– обрати конкретний метод, який буде ефективно підходити для інтеграції в компонент системи обліку пошкодженого житла;

– математично описати вхідні та вихідні змінні задачі дослідження.

Наступним кроком є розробка інформаційної технології, яка буде складатися з таких етапів як:

а) розробка функціональних вимог;

б) розробка моделі потоків даних;

в) розробка вимог для клієнтської частини технології:

1) розробка Use-Case діаграми;

2) розробка діаграми класів;

3) розробка діаграми послідовності.

г) опис архітектури;

ґ) фізичне моделювання даних;

д) імплементація нечіткої логіки у компонент пріоритизації системи;

е) тестування та проведення аналізу перспектив розвитку.

2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ, МЕТОДІВ, ЗАСОБІВ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ПРІОРИТЕЗАЦІЇ

2.1 Огляд підходів до вирішення задачі пріоритезації в науковій літературі

Проблема визначення пріоритетів відіграє ключову роль у багатьох сферах – від управління завданнями в системах реального часу до організації черговості в наданні медичної допомоги, гуманітарного реагування, розподілу обмежених ресурсів або формування черги на компенсації. Залежно від галузі застосування, застосовуються різні методи пріоритезації: від класичних формальних алгоритмів до стохастичних моделей та підходів на основі нечіткої логіки.

Одним із найбільш розроблених напрямів є призначення пріоритетів у комп’ютерних системах реального часу. В оглядовій статті «A Review of Priority Assignment in Real-Time Systems» автора R. I. Davis систематизовано сучасні алгоритми управління пріоритетами у вбудованих та критичних програмних системах [10]. Зокрема, розглядаються підходи з фіксованими пріоритетами, динамічними пріоритетами та алгоритми оптимального призначення пріоритетів. Основна особливість цих моделей полягає у суворій формалізації: усі вхідні параметри (час виконання, періодичність, дедлайни) вважаються точними та детермінованими. Завдяки цьому можливе математичне гарантування вчасного завершення задач. Проте подібні методи малопридатні для соціальних або гуманітарних задач, де критерії пріоритезації не завжди можна формалізувати кількісно.

Інший підхід представлено в дослідженні «Priority Assignment in Emergency Response» Evin Uzun Jacobson та співавторів, яке присвячене задачі пріоритезації у системах екстреного реагування [11]. У межах цього дослідження розроблено модель оптимального розподілу ресурсів за

допомогою динамічного програмування, де враховуються як важливість кожного запиту, так і обмеженість наявних ресурсів. Автори підкреслюють, що в умовах перевантаження система повинна віддавати перевагу не завжди найкритичнішим випадкам, а тим, які дають найбільший очікуваний результат. Такий підхід дозволяє оптимізувати загальну ефективність реагування, але потребує точних математичних моделей поведінки системи та прогнозування сценаріїв розвитку ситуації.

У сучасних задачах розподілу ресурсів усе частіше виникають ситуації, коли пріоритети не задані явно або є невизначеними. Наприклад, у роботі «Fairness in the Assignment Problem with Uncertain Priorities» [12] за авторством Zeyu Shen досліджується задача призначення за умов невизначених пріоритетів. Автори вводять поняття стохастичної справедливості та справедливості з урахуванням ймовірностей, які дозволяють будувати алгоритми розподілу ресурсів навіть у випадку, коли немає чіткої ієрархії між запитами. Такий підхід особливо цінний у задачах соціального спрямування – наприклад, у розподілі гуманітарної допомоги або формуванні черги на надання компенсацій, де важко визначити, чия потреба є «важливішою».

Узагальнюючи, можна стверджувати, що сучасна наукова література пропонує широкий спектр методів визначення пріоритетів. Алгоритмічні методи забезпечують високу ефективність і точність у формалізованих середовищах (системи реального часу, виробничі процеси), стохастичні – дозволяють працювати із ймовірнісною невизначеністю, а моделі соціальної справедливості – з невизначеною перевагою. Проте спільним недоліком багатьох класичних моделей є залежність від повноти та точності вхідної інформації, а також потреба у формалізованих критеріях пріоритету. У реальних системах, пов’язаних з гуманітарною допомогою, соціальними виплатами часто відсутній чіткий механізм прийняття рішень. Вхідні дані можуть бути неповними, суперечливими або представленими у вигляді якісних експертних оцінок. За таких умов актуальним постає застосування

підходів, здатних обробляти невизначеність і вербальні знання, зокрема – методів нечіткої логіки.

2.2 Доцільність застосування нечіткої логіки для задач пріоритезації в умовах невизначеності

В контексті умов, коли чіткий механізм прийняття рішень невідомий. особливу цінність набуває нечітка логіка, яка була спеціально розроблена для моделювання нечітких, неточних і суб'єктивних знань. На відміну від класичних формальних методів, нечітка логіка дозволяє працювати з лінгвістичними змінними (наприклад, «низький», «середній», «високий»), визначати для них функції належності та використовувати правила виду «ЯКЩО-ТО» для побудови висновків. У такий спосіб можна моделювати міркування експертів, які приймають рішення в реальних умовах на основі досвіду та якісних оцінок.

Ефективність застосування механізмів нечіткої логіки для задач гуманітарного реагування підтверджується результатами сучасних наукових досліджень. Зокрема, у статті «Fuzzy ontological model of knowledge representation for the humanitarian response» авторства Ольги Чалої, Тетяни Білової, Ірини Побіженко та Олени Остапенко [13] запропоновано нечітку онтологічну модель представлення знань для системи підтримки рішень у сфері гуманітарної допомоги. Модель поєднує онтологію предметної області, метод прецедентів та механізми нечіткого виведення. Це дозволяє, по-перше, структуровано описати знання про предметну область (типи надзвичайних ситуацій, ресурси, ризики, дії тощо), а по-друге – застосовувати нечіткі правила для модифікації рішень у новій ситуації з урахуванням невизначеності.

У межах цієї моделі ключові параметри, такі як «доступність ресурсу», «рівень ризику», «стан інфраструктури», описуються за допомогою лінгвістичних змінних та відповідних функцій належності. Наприклад,

значення «доступність ресурсу» може належати до термів «низька», «середня», «висока» з відповідними функціями належності на інтервалі [0,1]. На основі таких змінних будується база продукційних правил виду «ЯКЩО доступність низька І ризик високий, ТО необхідна евакуація». Застосування механізмів нечіткого виведення типу Мамдані дозволяє системі будувати адаптивні рішення, навіть за відсутності повної інформації. Автори доводять, що врахування нечітких співвідношень між поняттями (наприклад, між типом загрози та рівнем пошкодження, або між доступністю інфраструктури та ефективністю заходів) дозволяє істотно підвищити точність класифікації ситуацій: до 10-15% у порівнянні з класичними методами прецедентів.

Таким чином, аналіз сучасних наукових робіт свідчить про високу ефективність застосування нечіткої логіки для задач пріоритезації в умовах невизначеності, багатофакторності та відсутності чітких правил. На відміну від класичних формалізованих підходів, нечітка логіка дозволяє враховувати досвід експертів, працювати з якісними оцінками та будувати гнучкі адаптивні рішення. Це робить її особливо придатною для побудови моделей пріоритезації у сфері компенсацій за пошкоджене житло, розподілу гуманітарної допомоги та інших задач соціального спрямування, де важливо забезпечити справедливість, адаптивність і обґрунтованість рішень у складних та нестабільних умовах.

2.3 Опис системи нечіткого виведення

У процесі дослідження методів визначення пріоритету виплат було встановлено, що значна частина вхідних даних є нечіткою, має соціальний або якісний характер і не може бути виражена строго у числовому вигляді. Саме тому для побудови алгоритму пріоритезації доцільно використати методи нечіткої логіки, які дозволяють поєднати кількісні й якісні оцінки параметрів у єдиній системі логічного висновку.

Визначено вхідні параметри системи нечіткого виведення. У компоненті пріоритизації використано п’ять вхідних змінних:

– destruction: ступінь пошкодження житла;

– residents: кількість зареєстрованих мешканців;

– age: вік власника житла;

– disability_group: група інвалідності (0 – інвалідність відсутня, 1 – важка ступінь, 2 – середня ступінь, 3 – легка ступінь);

– is_large_family — ознака багатодітної сім’ї (0 або 1).

Вихідна змінною є priority (пріоритет): чим більше число, тим вище пріоритетність виплати.

Для обчислення ступеня пошкодженого житла потрібно мати наступні вхідні дані для всього чек-листу:

– criticality: критичність елементу чек-листу для житла, значення коефіцієнту може лежати в інтервалі [0.1, 1] (наприклад для ремонту несучих стін criticality = 1, для заміни вікон 0.4, а для поклейки шпалер 0.1, тощо);

– damage_level: рівень пошкоджень, значення коефіцієнту може лежати в інтервалі [0.1, 1], допустимі значення (low = 0.3, medium = 0.6, high = 1).

Виходячи з вхідних змінних, отримуємо формулу для обчислення рівню пошкодження:

$$desctruction = 10\frac{\sum criticality_{\,i} * damage\_level}{\sum criticality_{\,i}},$$

де коефіцієнт 10 нормує значення вихідної змінної destruction в інтервал від 1 до 10.

Наступним етапом є побудова функцій належності (ФН). Вибір трикутних ФН є доцільним у ситуаціях, коли можна чітко виділити одне конкретне ідеальне значення, яке найкраще описує лінгвістичний термін. Така форма підходить для випадків, де поняття має виражений пік і швидке

спадання істинності при найменшому відхиленні від цього центру, що забезпечує високу чутливість системи до змін вхідних даних.

Натомість трапецієподібні функції слід застосовувати тоді, коли поняття передбачає певний інтервал значень, які є рівнозначно правильними або істинними. Наявність верхнього плато дозволяє системі ігнорувати незначні коливання вхідного сигналу в межах цього ядра, забезпечуючи більшу стабільність і толерантність до шуму.

Вхідна змінна destruction (ступінь пошкодження житла) має діапазон допустимих значень від 1 до 10 та трапецевидні ФН, які наведено на рис. 2.1.

Терми змінної:

– low (1, 1, 2.5, 4);

– medium (3, 4.5, 5.5, 7);

– high (6, 7.5, 10, 10).

Такі інтервали забезпечують плавний перехід між рівнями пошкодження – від незначного до критичного.

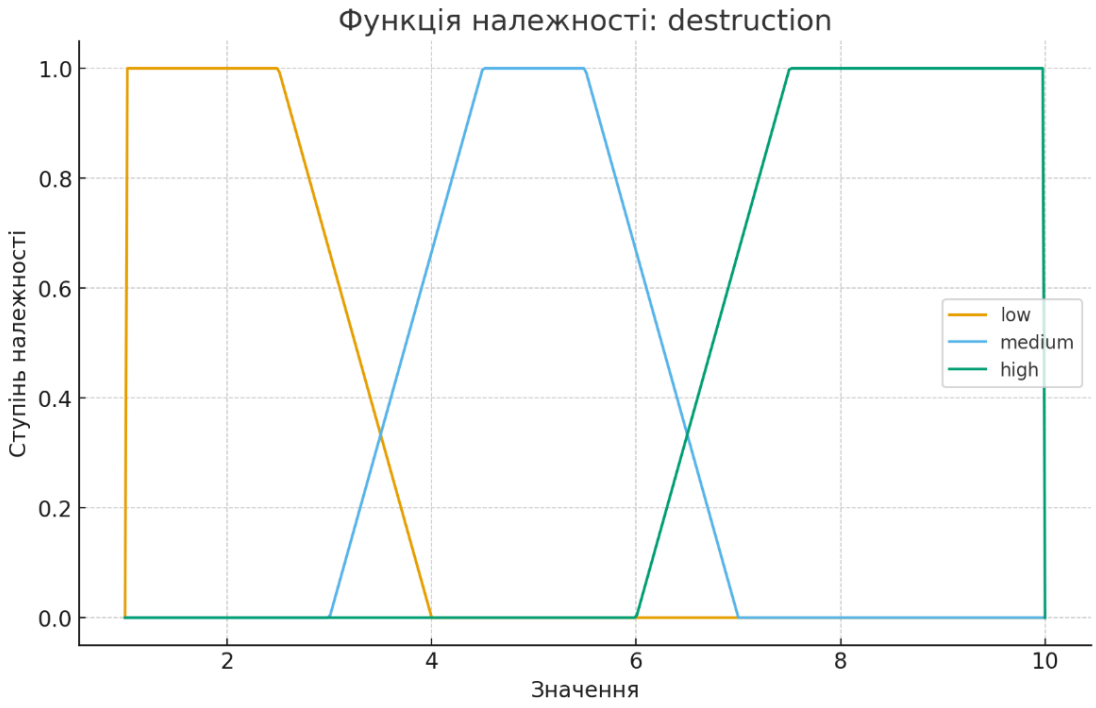

Рисунок 2.1 – Функція приналежності desctruction

Вхідна змінна residents має діапазон допустимих значень від 1 до 10 та трапецевидні ФН, які зображено на рис. 2.2. Терми змінної:

– single (один) (1, 1, 1.5, 2);

– normal (1.5, 2.5, 4.5, 5.5);

– many (4.5, 6, 10, 10).

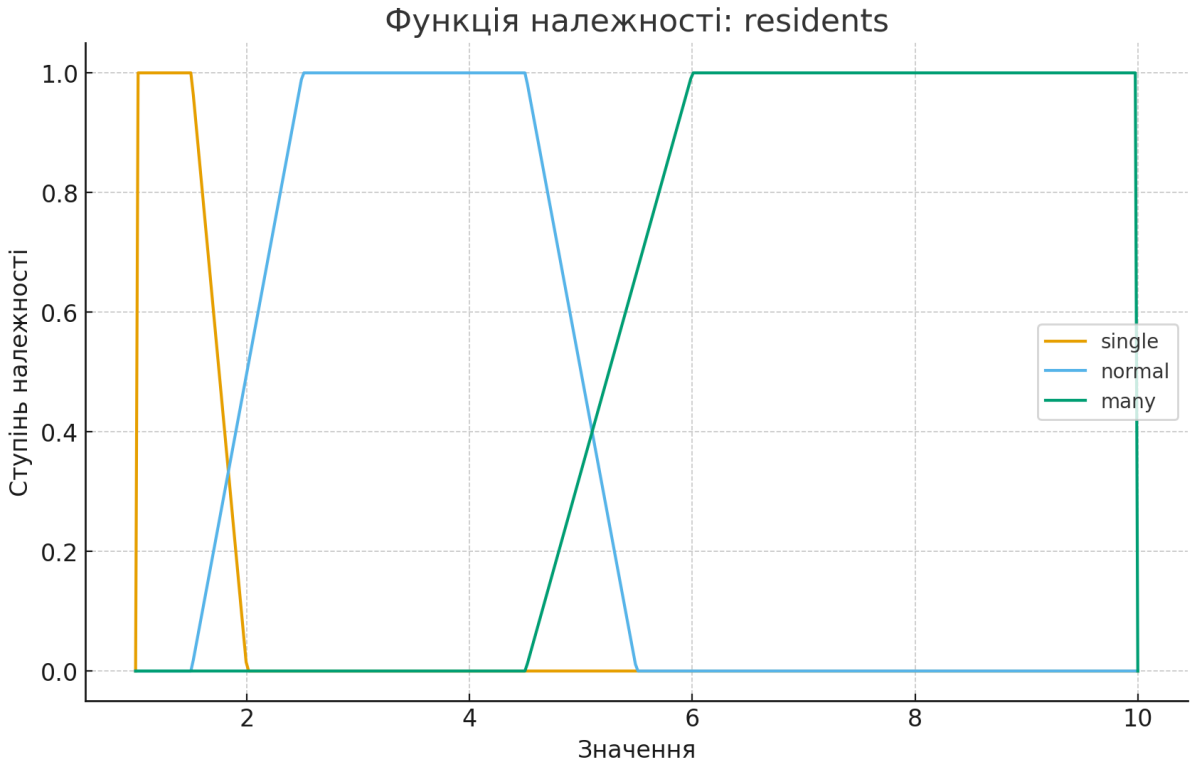

Рисунок 2.2 – Функція приналежності residents

Вхідна змінна age має діапазон допустимих значень від 18 до 100 та трапецевидні ФН, які зображено на рис. 2.3. Терми змінної :

– young (18, 18, 28, 35);

– middle (32, 38, 50, 60);

– elderly (58, 70, 100, 100).

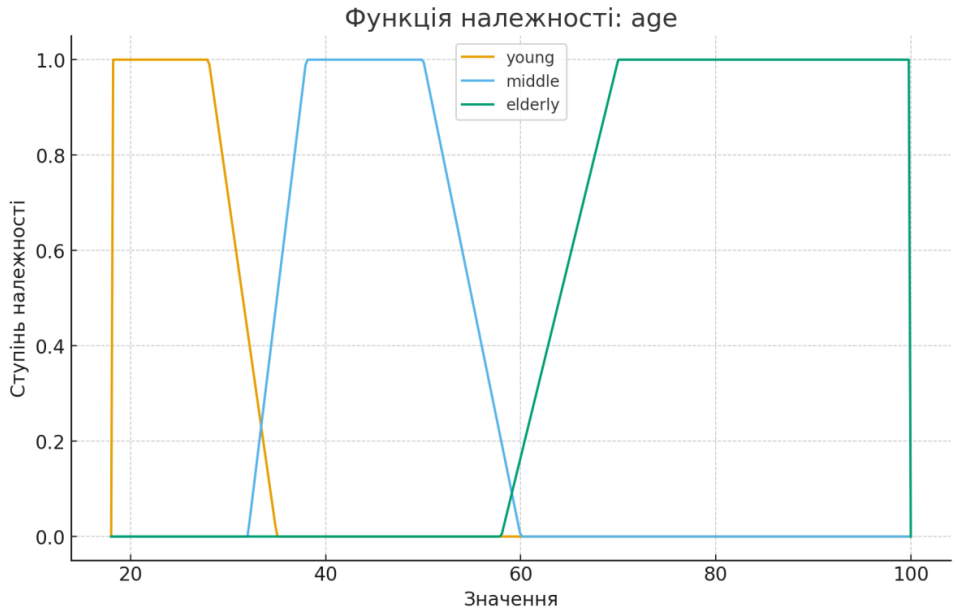

Рисунок 2.3 – Функція приналежності age

Вхідна змінна disability_group має діапазон допустимих значень від 0 до 3 та трикутні ФН, які зображено на рис. 2.4. Терми змінної:

– none (0, 0, 1);

– severe (0, 1, 2);

– medium (1, 2, 3);

– light (2, 3, 3).

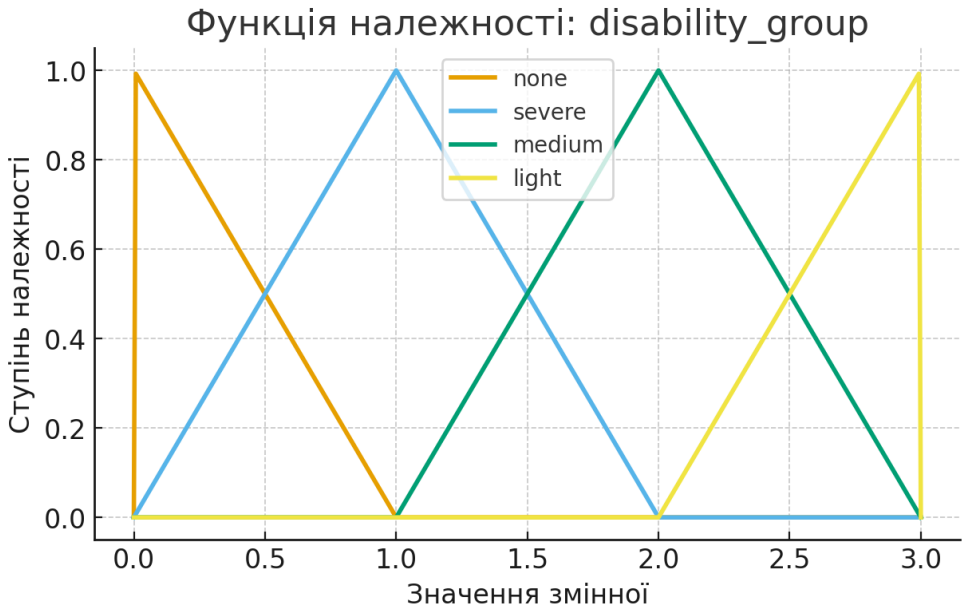

Рисунок 2.4 – Функція приналежності disability_group

Вхідна змінна is_large_family має діапазон допустимих значень від 0 до 1 та трикутні ФН, які зображено на рис. 2.5. Терми змінної:

– no (0, 0, 1);

– yes (0, 1, 1).

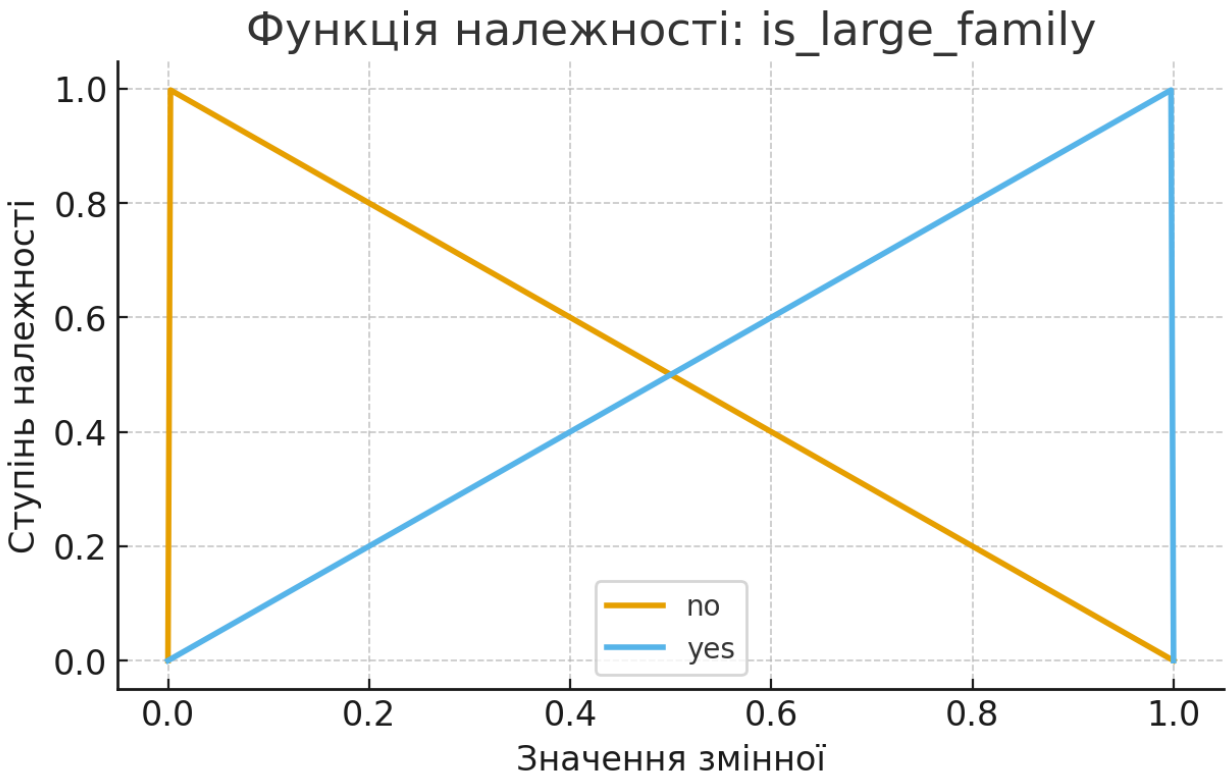

Рисунок 2.5 – Функція приналежності is_large_family

Вихідна змінна priority має діапазон допустимих значень від 0 до 10. Терми змінної:

– low (Низький): (1, 1, 2, 4)

– medium (Середній): (3, 5, 6, 8)

– high (Високий): (7, 9, 10, 10)

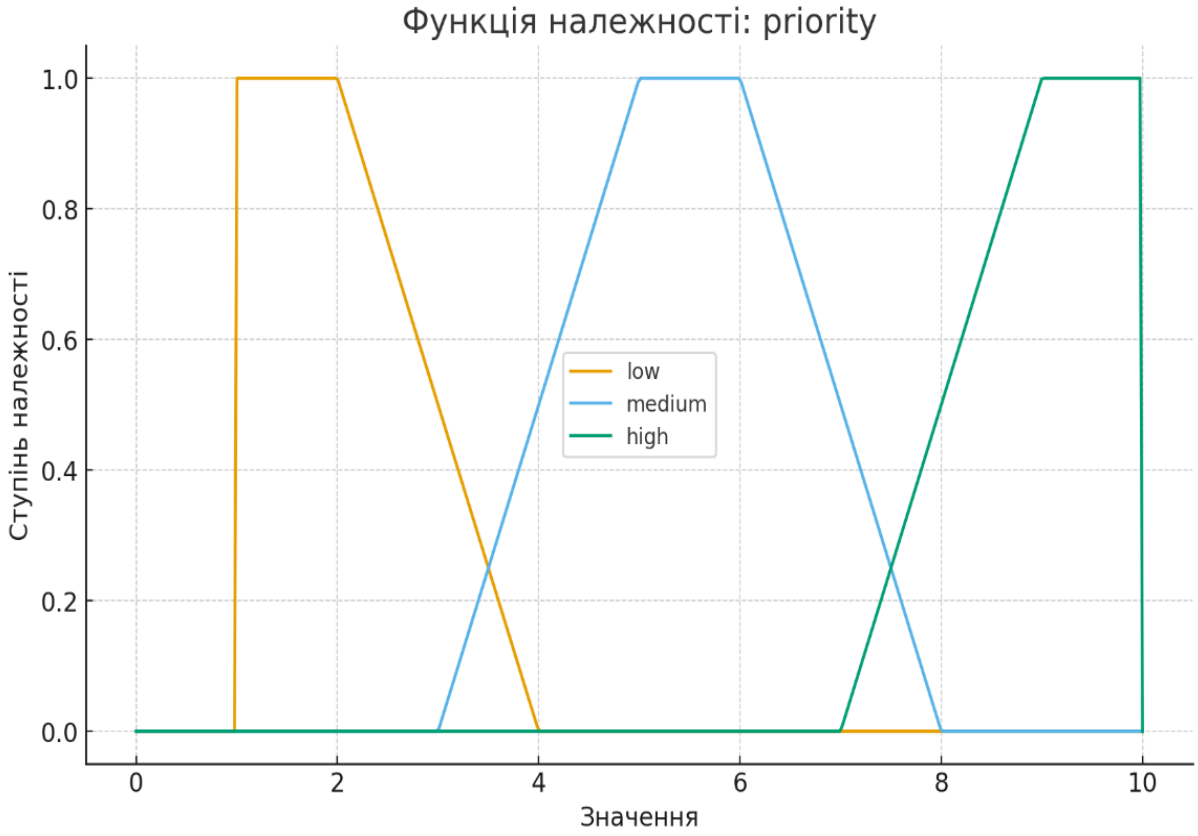

Рисунок 2.6 – Функція приналежності priority

2.4 Порівняння систем Мамдані та Сугено

Визначившись з вхідними та вихідними параметрами потрібно порівняти Система Мамдані базується на нечітких множинах як для вхідних, так і для вихідних змінних. Усі проміжні етапи висновку залишаються у нечіткій формі, а кінцевий результат перетворюється у числове значення лише після дефаззифікації. Математично висновок описується правилами:

якщо x_1 це A_1^k та x_2 це A_2^k, то y це B^k, де A_i^k – нечіткі множини вхідних змінних, а B^k – нечітка множина виходу.

У системі Сугено вихід не є нечіткою множиною, а подається як аналітична функція вхідних змінних:

$$y_k = a_0 + a_1x_1 + a_2x_2 + \ldots + a_nx_n.$$

Остаточний результат визначається як зважене середнє за ступенями належності кожного правила:

$$y = \frac{\sum_k w_k y_k}{\sum_k w_k}.$$

Порівняльний аналіз показує, що модель Мамдані краще підходить для соціально-гуманітарних систем, оскільки її результати є інтерпретованими у природній формі та легко пояснюваними користувачу. Модель Сугено забезпечує більшу швидкодію і точність при наявності великої кількості даних, проте втрачає прозорість і потребує навчання для налаштування параметрів функцій [14].

Для компоненту обрано систему Мамдані, оскільки вона забезпечує гнучкість, зрозумілість логіки висновку та можливість опису правил безпосередньо в термінах, зрозумілих експертам-бухгалтерам і адміністраторам фондів.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИРІШЕННЯ ЗАДАЧІ

3.1 Розробка функціональних вимог до технології пріоритезації

Для розробки функціональних вимог використано ПЗ ALLFusion Process Modeler яке спеціалізоване для моделювання бізнес-процесів. Воно дозволяє візуалізувати, аналізувати і оптимізувати бізнес-процеси організації за допомогою створення діаграм. Цей інструмент допомагає покращити управління процесами, підвищити їх ефективність та полегшити прийняття рішень, надаючи засоби для детального документування і аналізу поточних та цільових процесів [15].

На контекстній діаграмі IDEF0 (рисунок 3.1) зображено загальну структуру роботи інформаційної технології пріоритезації виплат системи обліку пошкодженого житла. Основним процесом є блок «Компонент пріоритизації виплат системи обліку пошкодженого житла», який виконує обчислення пріоритету для кожної заявки та формує фінальний список виплат. На вхід системи надходять три основні потоки даних: «Дані про житло», «Дані про власника житла» та «Заповнений чек-лист пошкоджень», що передаються з основної інформаційної системи (ІС).

Керуванням виступають «Нормативна документація», яка визначає правила розподілу коштів, і «Керівництво користувача», яке регламентує дії бухгалтера під час формування списків. Механізмами виконання процесу є ІС, яка забезпечує зберігання та обробку даних, і бухгалтер, який ініціює обчислення та виконує перегляд результатів. На виході з компоненту формуються два потоки: «Пріоритет виплати» числовий коефіцієнт, що відображає черговість компенсації, і «Виплата» – кінцеве рішення про здійснення компенсації постраждалому власнику житла.

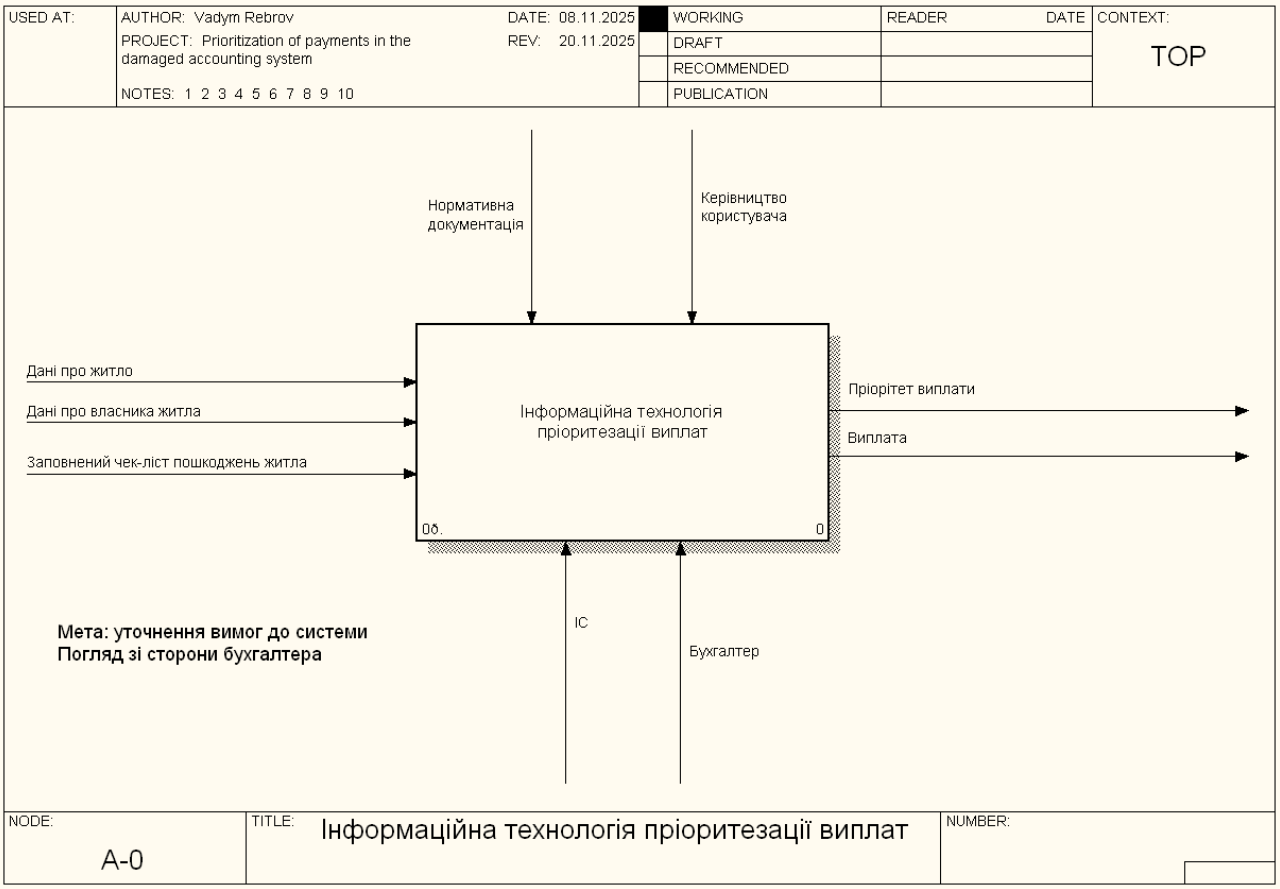

Рисунок 3.1 – Контекстна діаграма

Діаграма декомпозиції, яка зображена на рис. 3.2, розкриває внутрішню структуру компоненту пріоритизації виплат і деталізує три основні підпроцеси. Перший процес «Обчислення рівня пошкодження житла» – отримує заповнений чек-лист пошкоджень і перетворює його у числову оцінку рівня руйнування будівлі. Результатом цього процесу є показник «Рівень пошкодження житла», який передається до наступного блоку.

Другий процес «Обчислення пріоритету виплати» – є центральним. Він приймає на вхід рівень пошкодження, дані про житло та дані про власника, зокрема вік, кількість мешканців, групу інвалідності й наявність багатодітної сім’ї. На основі цих параметрів система формує інтегральний показник пріоритету. Отриманий коефіцієнт визначає відносну важливість кожної заявки. Третій процес «Створення списку виплат» – об’єднує обчислені пріоритети, сортує заявки у порядку спадання значень і формує структурований список для подальшої обробки бухгалтером.

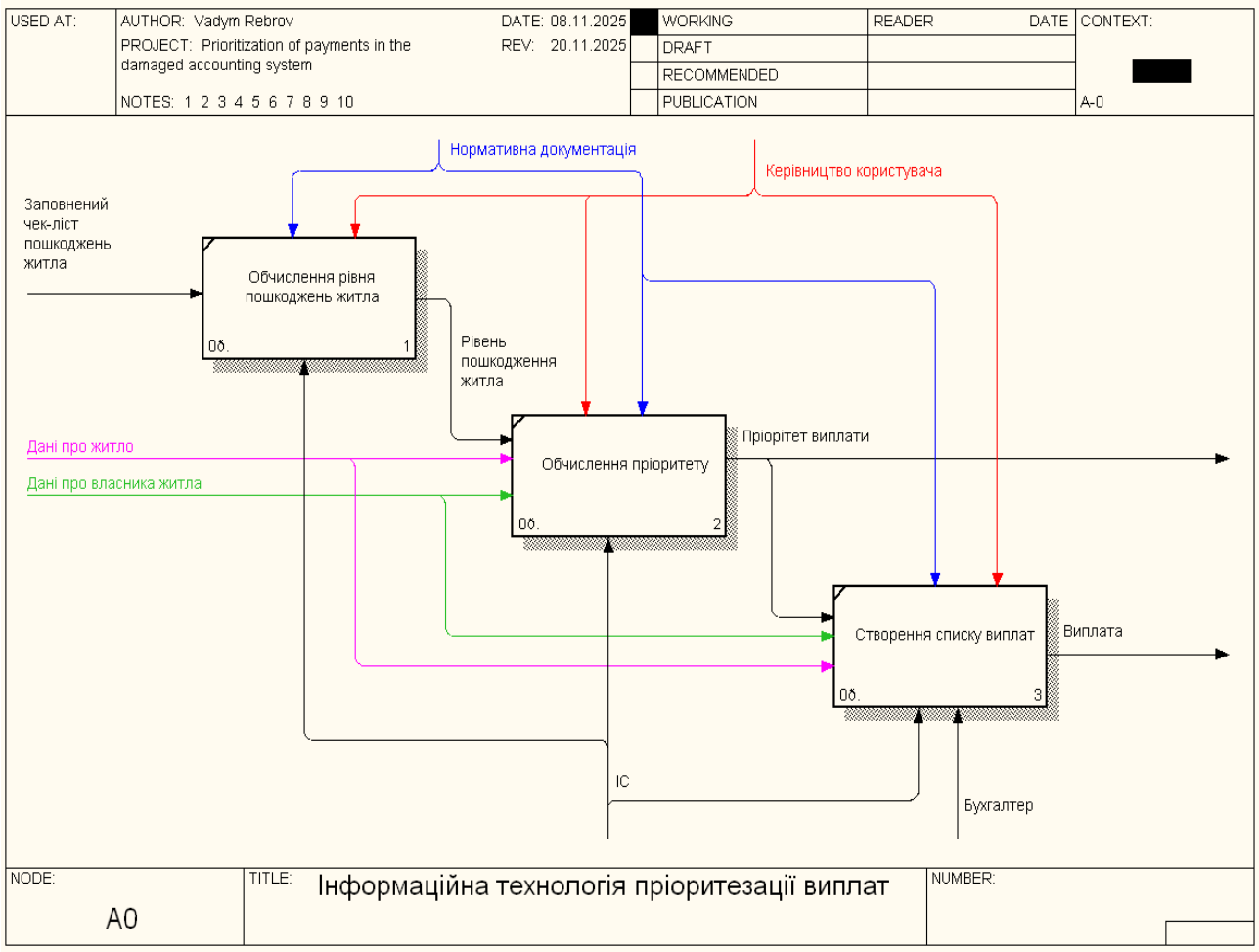

Рисунок 3.2 – Діаграма декомпозиції

Третій процес «Створення списку виплат» – об’єднує обчислені пріоритети, сортує заявки у порядку спадання значень і формує структурований список для подальшої обробки бухгалтером. На виході формується потік «Виплата», який може бути збережений у базі даних або експортований у вигляді PDF-звіту.

Між процесами передаються інформаційні потоки: рівень пошкодження, дані про житло, дані про власника, пріоритет та фінальна інформація про виплати.

Таким чином, діаграма декомпозиції IDEF0 демонструє логічну послідовність обробки даних від моменту отримання результатів огляду житла до формування кінцевих фінансових списків, що передаються до бухгалтерії.

3.2 Розробка моделі потоків даних технології пріоритезації

Контекстна діаграма DFD, яка зображена на рис. 3.3, описує основні потоки даних між зовнішніми сутностями та компонентом пріоритизації виплат.

Головним зовнішнім джерелом інформації є інформаційна система (ІС), що надає три потоки даних: «Дані про житло», «Дані про власника житла» та «Заповнений чек-лист пошкоджень». Ці дані передаються до процесу «Компонент пріоритизації виплат», який виконує обчислення відповідно до закладених алгоритмів нечіткої логіки.

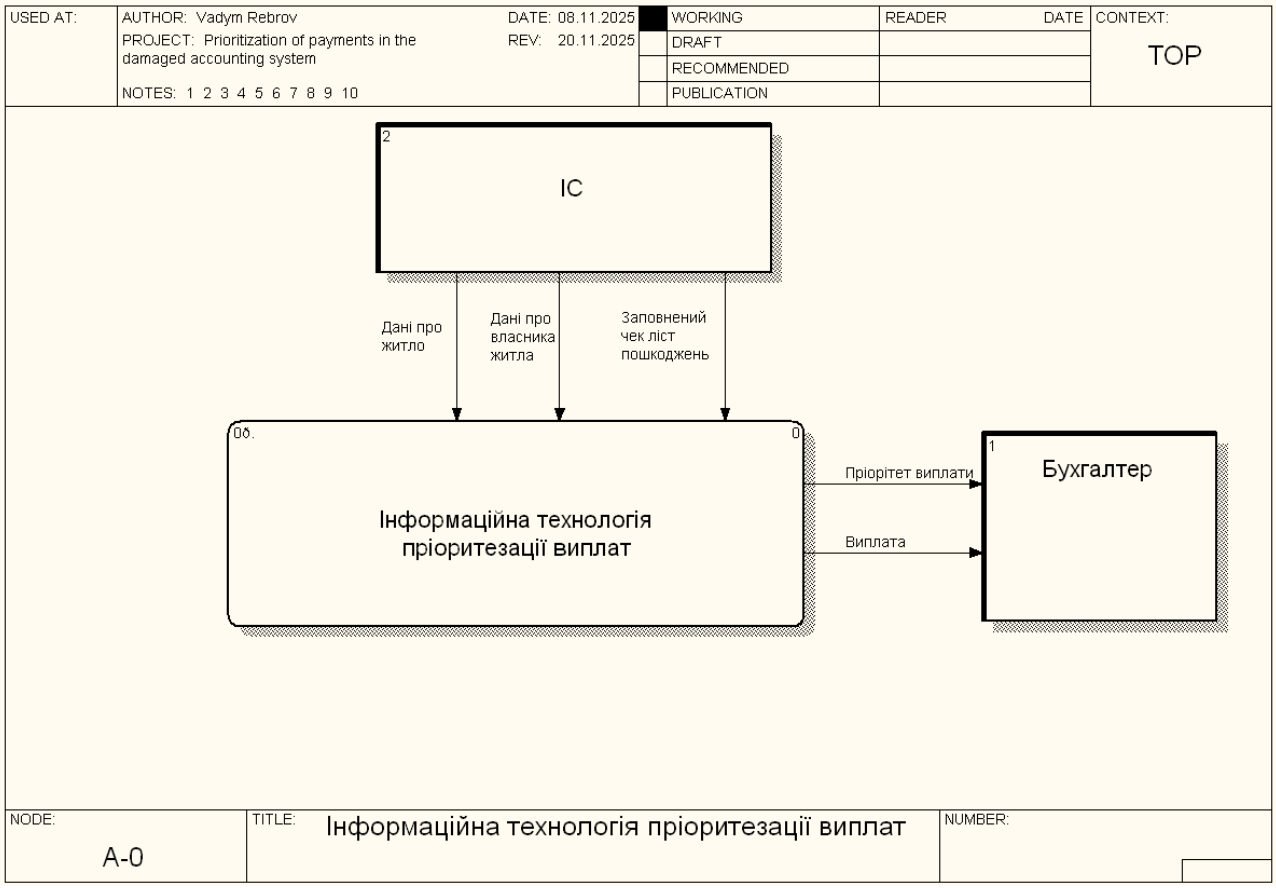

Рисунок 3.3 – Контекстна діаграма

Отримані результати у вигляді «Пріоритету виплати» та «Виплати» передаються зовнішній сутності – бухгалтеру, який виконує перевірку, затвердження або коригування сформованих списків. Таким чином, на

діаграмі видно два зовнішні об’єкти: ІС як джерело вхідних даних і бухгалтер як кінцевий користувач, який отримує результати роботи компоненту. Декомпозиція DFD деталізує структуру процесів усередині компоненту пріоритизації виплат, результат якої наведено на рис. 3.4.

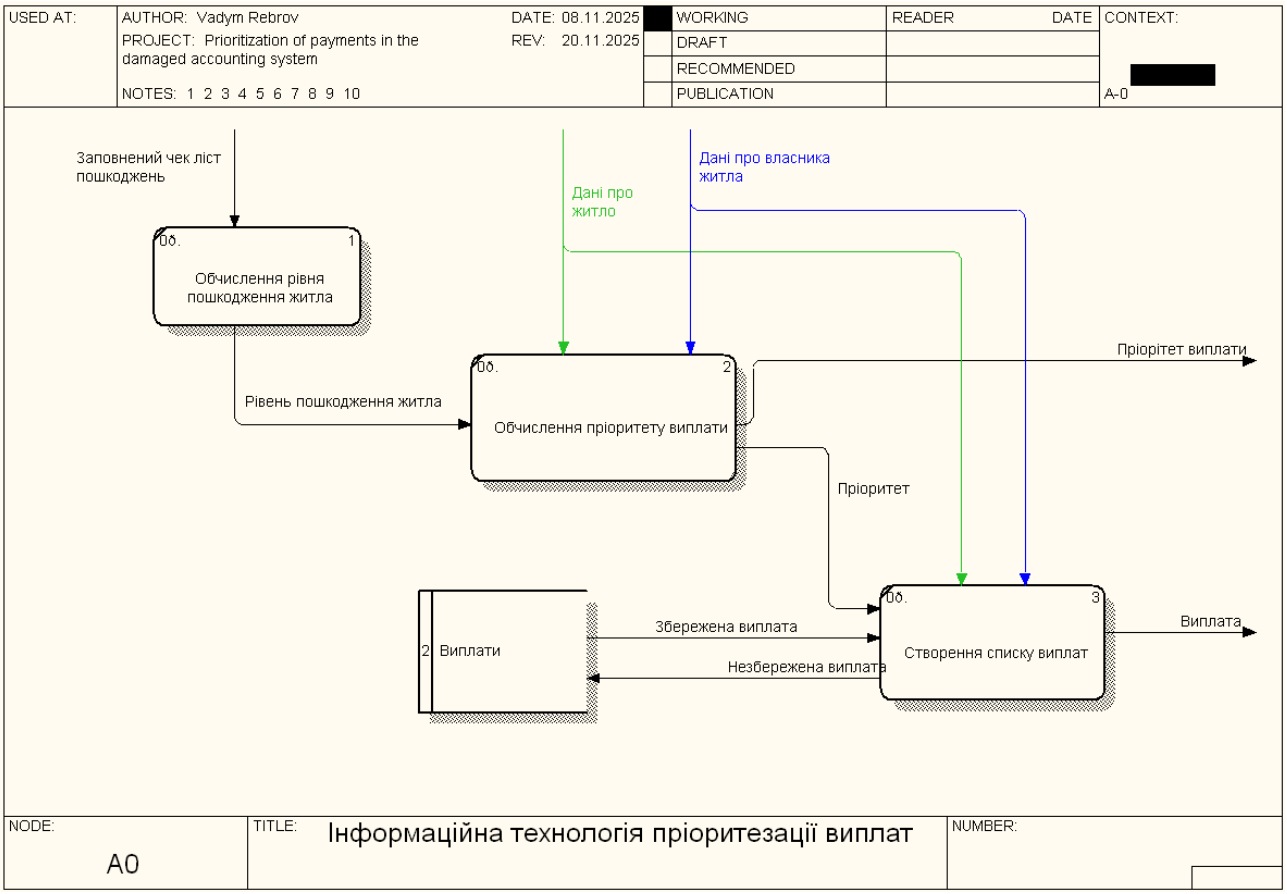

Рисунок 3.4 –Діаграма декомпозиції

Перший процес «Обчислення рівня пошкодження житла» отримує вхідний потік «Заповнений чек-лист пошкоджень» і перетворює його у цифровий показник рівня пошкодження. Другий процес «Обчислення пріоритету виплати» використовує результати першого процесу разом із даними про житло та власника. Вихідним потоком цього процесу є «Пріоритет виплати». Далі процес «Створення списку виплат» об’єднує всі отримані пріоритети, перевіряє наявність збережених даних, сортує заявки й формує кінцевий вихідний потік «Виплата». Потоки між процесами забезпечують

передачу рівня пошкодження, даних про житло та власника, а також пріоритету як проміжного результату.

3.3 Розробка вимог до інтерфейсу клієнтської частини технології пріоритезації

Відповідно до стандарту ISO29110 [16] створено відповідності C-вимог (вимоги процесу) та D-вимог (вимоги до продукту), які наведено у таблиці 3.1

Таблиця 3.1 – Вимоги до продукту

Актор	Задача	Номер C-вимоги та її формулювання	Номер D-вимоги та її формулювання
Бухгалтер	1. Управління списком виплат	1.1 Бухгалтер повинен мати можливість переглянути веб сторінку з автоматично згенерованим списком виплат у вигляді таблиці, відповідно до обраного населеного пункту.	1.1.1 Система повинна надавати список виплат у форматі таблиці, що містить наступні дані: номер пріорітету, інформація про житло, власника, його номер IBAN, сума виплати. 1.1.2 Система повинна надавани можливість переглядати відфільтровані списки за областями. 1.1.3 Система повинна формувати списки для виплат виключно з того переліку житла, що має статус у системі «Approved» (огляд житла підтверджено) і чек-ліст житла містить 1 або більше елементів. 1.1.4 Система повинна виконувати пошук за створеним індексом адреси у базі даних
		1.2 Бухгалтер повинен мати можливість шукати житло у списку виплат за номером телефону власника	1.2.1 Система повинна надавани можливість переглядати відфільтровані списки за областями.
		1.3 Бухгалтер повинен мати можливість завантажити список виплат у форматі PDF	1.3.1 Система повинна формувати список виплат у форматі PDF відповідно до заданих фільтрів. 1.3.2 Список виплат у PDF повинен містити всі релевантні дані: номер пріоритету, інформація про житло, власника, його номер IBAN, сума виплати.

3.3.1 Розробка діаграми варіантів використання технології пріоритезації

Для розробки вимог для технології використано UML-моделювання, яке призначене для візуалізації, специфікації, конструювання та документування компонентів програмних систем. Вони допомагають розробникам і аналітикам описувати структуру і поведінку. Використання UML діаграм сприяє більш ефективному проектуванню систем, покращує комунікацію між розробниками та замовниками, а також допомагає знаходити і виправляти помилки на ранніх етапах розробки [17].

На рисунку 3.5 наведено Use-case діаграма компоненту системи. Вона містить актора бухгалтера. Використовуються точки розширення «Отримати PDF-звіт) та включення до «Переглянути відфільтрований список» до прецеденту «Переглянути список пріоритизованих виплат».

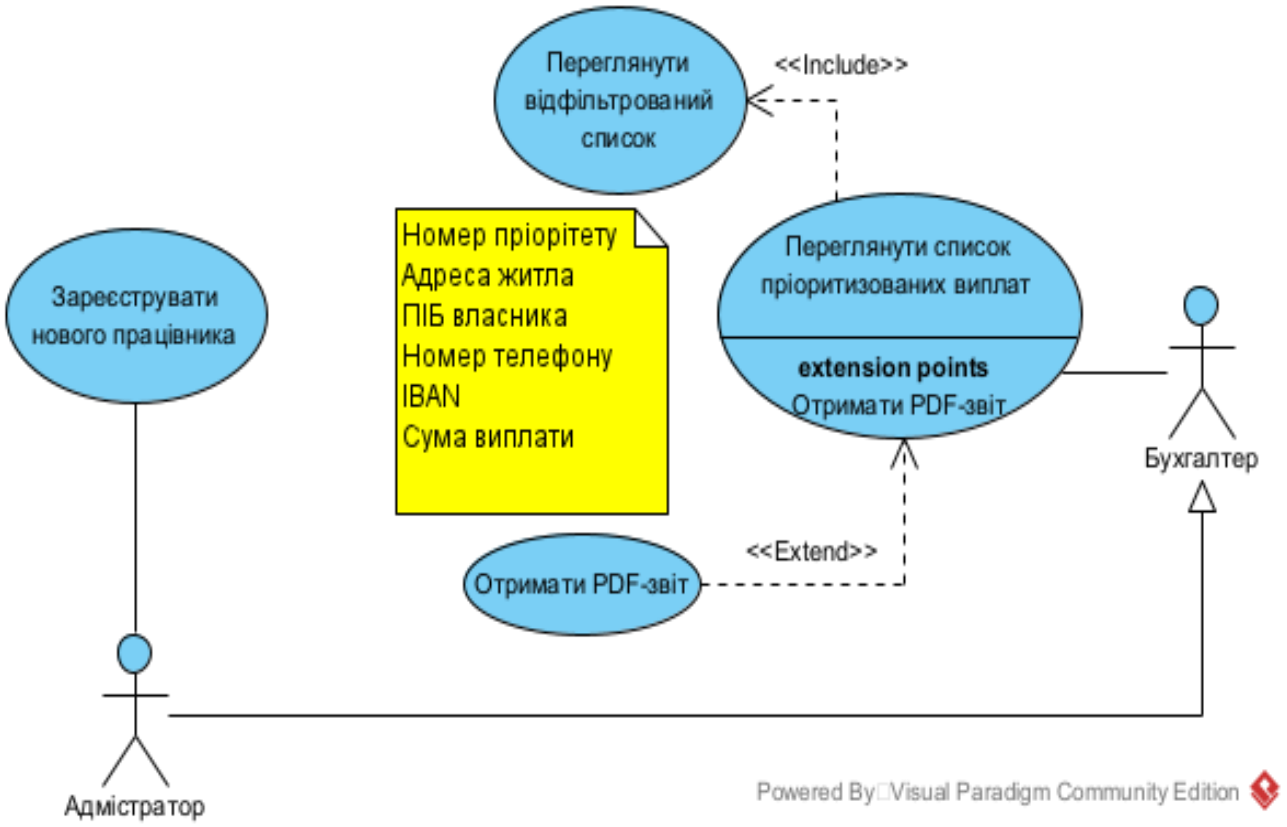

Рисунок 3.5 – Use-case діаграма ІТ пріоритезації виплат

3.3.2 Розробка діаграми класів технології пріоритезації

Для розроблюварого компоненту приоритезації створено діаграму класів (рис. 3.6). Наведена діаграма містить наступні класи-моделі: Address, Dwelling, DwellingOwner, DisabilityGroup, Payment, PaymentStatus. Для доступу до реалізованої бази даних використовується інтерфейс-репозиторій PaymentRepository.

Перелічені інтерфейси наслідковують функціонал від Spring CrudRepository, який містить методи: add, update, delete, get тощо.

Для того щоб зробити систему більш модульною, потрібно ізолювати бізнес-логіку, використовуючи класи сервіси, які зв’язані з відповідними класами репозиторіями відношенням агрегації – таким класом є PaymentService.

Для декларації HTTP запитів, за допомогою, яких можно звертатися до серверної частини системи, використовується контролер PaymentController.

Клас Dwelling представляє інформацію про житло, яке було пошкоджене та пройшло процес оцінки комісією. Об’єкт містить атрибути, що характеризують місцезнаходження житла (композиція з класом Address), кількість зареєстрованих осіб та рівень пошкодження. Атрибут destructionLevel є числовим показником, який згодом використовується у нечіткій моделі для визначення пріоритету виплати.

Клас DwellingOwner містить дані власника пошкодженого житла – ідентифікаційну інформацію, контактні дані та банківські реквізити IBAN, необхідні для здійснення компенсації. Окремо зберігається параметр disabilityGroup, що дозволяє класифікувати власників за групою інвалідності.

Клас Payment представляє фінансову операцію, яка формується для кожної заявки. Атрибут priorityNumber містить значення пріоритету, отримане в результаті нечіткого моделювання.

Сума компенсації зберігається в полі compensationSum, а статус виплати описано через перелік PaymentStatus, що передбачає три етапи: «не сплачено», «в процесі» та «сплачено».

Клас PriorityCalculator реалізує механізм визначення пріоритету за допомогою нечіткої логіки.

Він приймає об’єкти Dwelling та DwellingOwner, обчислює значення на основі п’яти факторів, застосовує відповідні функції належності та повертає числову оцінку пріоритету. Це – окремий сервісний клас, який не зберігає стан, а реалізує алгоритмічну частину компоненту.

Клас PaymentService виконує роль бізнес-логіки для роботи з виплатами. Він оперує репозиторієм PaymentRepository, який, у свою чергу, успадковує інтерфейс CrudRepository і відповідає за створення, оновлення та отримання даних зі сховища.

Сервіс також використовує PriorityCalculator для обчислення порядку виплат, а також реалізує функцію формування PDF-звіту.

3.3.3 Розробка діаграми послідовності дій технології пріорітезації

На рисунку 3.7 наведено Sequence діаграму процессу створення виплати. Вона містить основного актора Accountant (Бухгалтер), який призначає фінальну суму компенсації, яка передається у Dwelling Controller (головна система).

Далі сума компенсації переходить у DwellingService, де активується механізм створення виплати. Оновлені дані про житло та дані про власника передаються у компонент пріоритизації шляхом виклику ендпоінту PaymentController.

Далі дані переходять у PaymentService де викликається PriorityCalculator, який за допомогою нечіткої логіки визначає пріорітет виплати. Після цього пріоритизована виплата зберігається у БД та вся інформація про виплату передається користувачу «Бухгалтер» у форматі View.

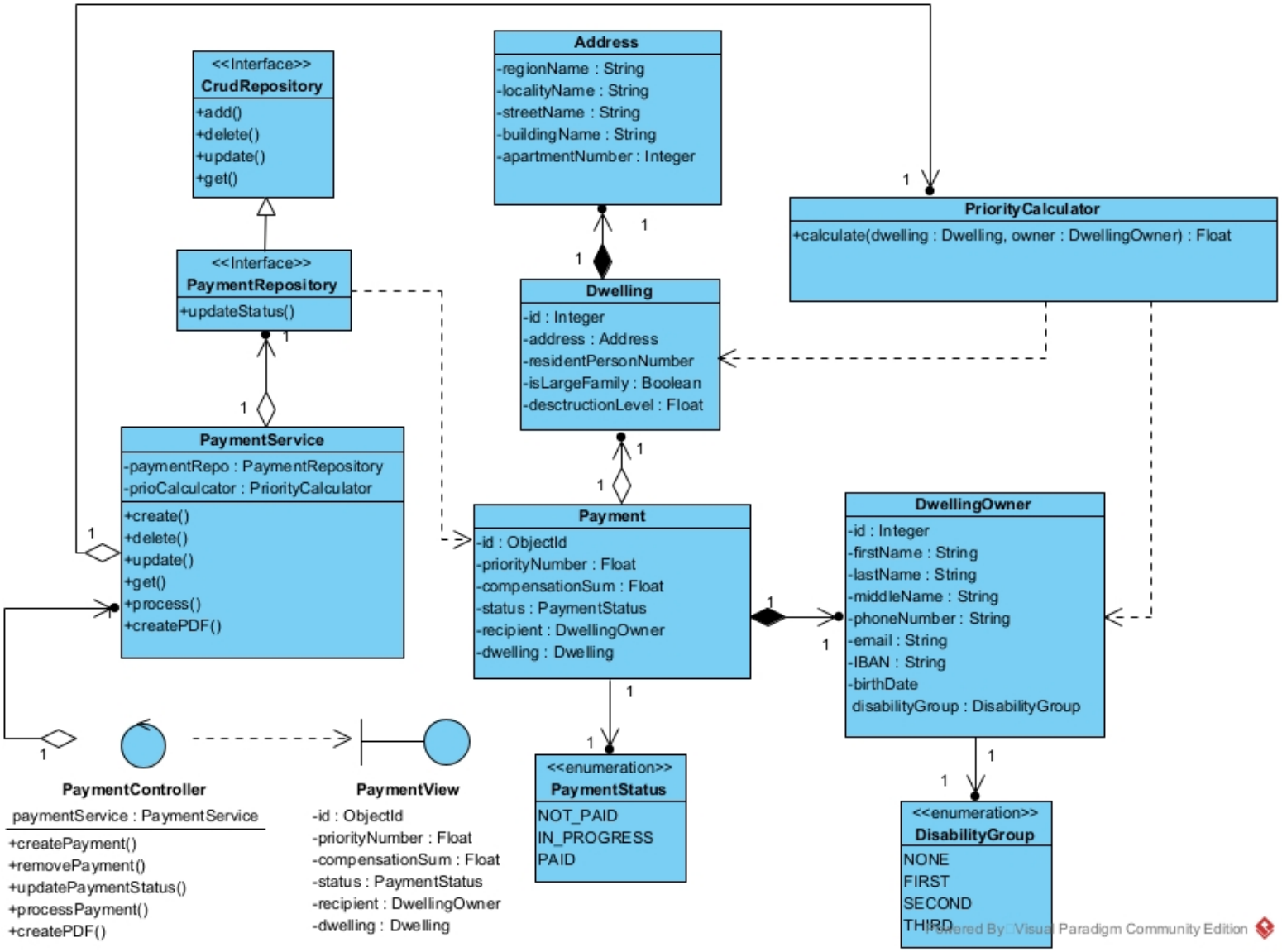

Рисунок 3.6 – Діаграма класів технології пріоритезації виплат

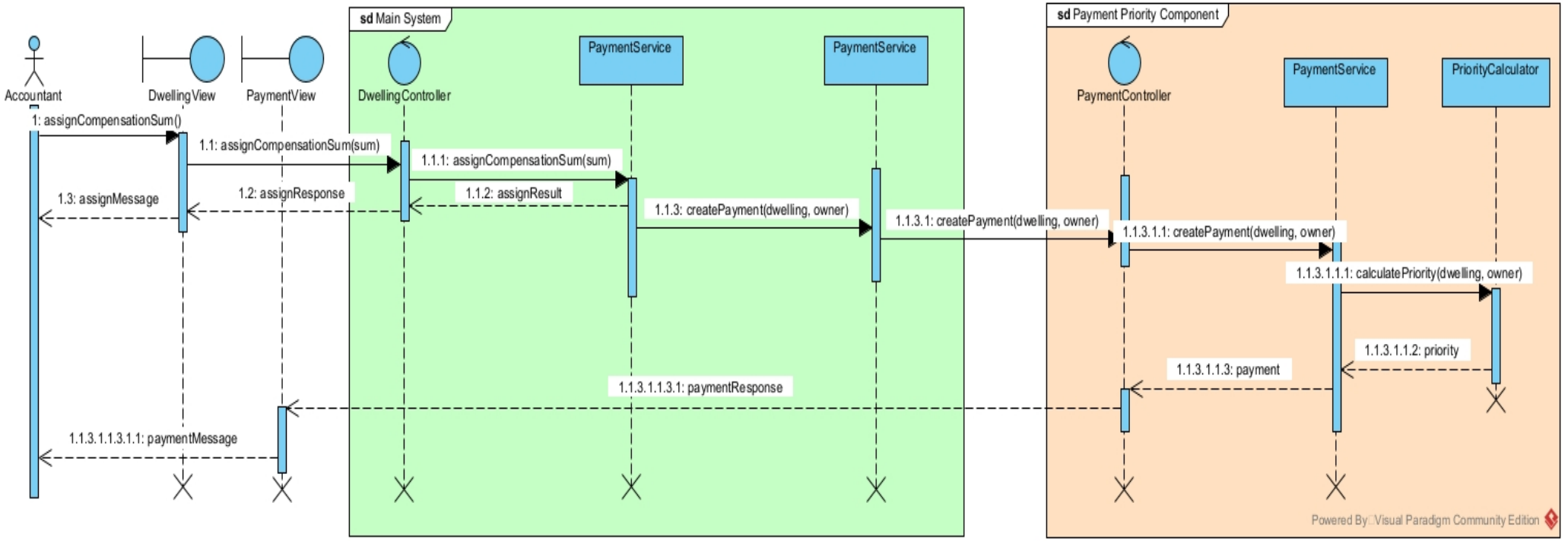

Рисунок 3.7 – Діаграма послідовностей для створення пріоритетної виплати

3.4 Опис архітектури розробленої системи

Архітектуру компоненту, який є мікросервісом, наведено на рис. 3.8, де зображено основну систему, яка взаємодіє з компонентом пріоритезації шляхом HTTP-запитів. Оскільки це окремий процес, його можна незалежно масштабувати або оновлювати, а збій у ньому не призведе до зупинки всієї системи обліку [18]. Сервіс виконує єдину спеціалізовану бізнес-функцію: автоматичне визначення пріоритету виплат – і має власний стек технологій: компонент побудований на Java Spring Framework [19] та використовує REST-контролери для створення API.

Для забезпечення виконання алгортимічної частини з використанням нечіткої логіки використано бібліотеку JFuzzyLite [20], яка надає зручний та оптимізований набір методів для нечіткого виведення типу Мамдамі та Сугено. Для збереження даних про виплати використано документоорієнтовану СУБД MongoDB, яка забезпечує високу швидкодію у читанні та запису даних про виплати [21], що критично для поставленої задачі. За відображення даних у клієнтській частині відповідає Thymeleaf [22].

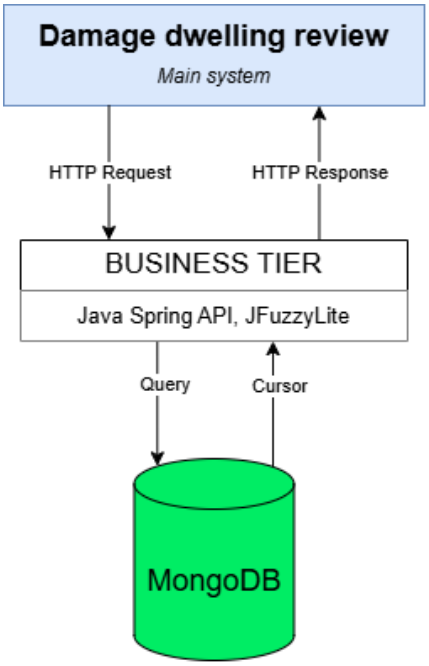

Рисунок 3.8 – Архітектура компоненту пріоритезації системи обліку пошкодженого житла

3.5 Фізичне моделювання даних системи

Оскільки для сховища даних про виплати обрано MongoDB, потрібно спроектувати структуру колекції. Для цієї задачі обрано Hackolade – це професійний інструмент для візуального моделювання даних, розроблений насамперед для NoSQL баз, але який зараз підтримує також SQL та API [23].

Він перетворює складні, вкладені JSON-структури на зрозумілі графічні схеми (дерева), дозволяючи проектувати базу даних візуально, без написання коду.

Основні переваги:

– на відміну від класичних інструментів, він ідеально відображає ієрархію документів NoSQL (масиви, об'єкти всередині об'єктів).

– reverse Engineering (візуалізація схеми даних вже існуючої БД);

– генерація скрипту валідацій MongoDB Validator.

На рис. 3.9 наведено фізичну модель даних.

Ідентифікатором документу є _id типу ObjectId – 12 байтне унікальне значення.

Атрибут priority містить числове значення пріоритету виплати.

Атрибут compensationSum містить суму виплати.

Вкладений документ dwellingOwner містить усю персональну інформацію про власника житла.

Вкладений документ dwelling містить повну адресу житла, рівень пошкодження житла, інформацію про багатодітність сім’ї що там проживає та кількість зареєстрованих мешканців.

Масив statuses містить інформацію про те коли змінювались статуси виплати.

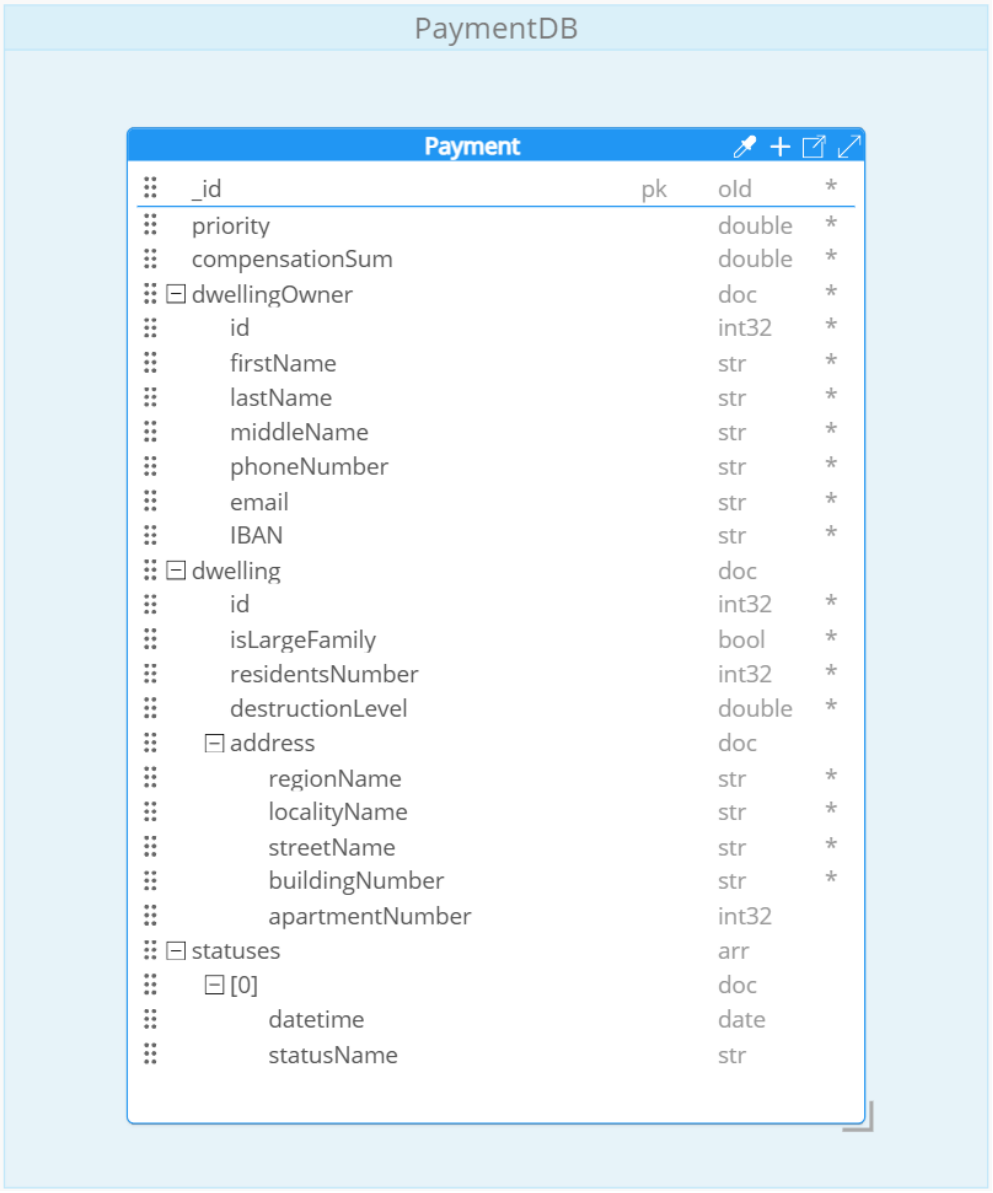

Рисунок 3.9 – Структура колекції виплат у БД

3.6 Імплементація нечіткої логіки у технологію пріоритезації

Для обчислення вихідної змінної у системі нечіткого виведення типу Мамдамі використовуючи бібліотеку JFuzzyLite потрібно створити файл написаний спеціальною мовою програмування – FCL (Fuzzy Control Language). Вона розроблена для формального опису систем нечіткої логіки і є частиною міжнародного промислового стандарту IEC 61131-7, який регламентує програмування нечіткого управління в програмованих логічних контролерах (ПЛК) та інших обчислювальних системах [24].Основною метою впровадження стандарту IEC 61131-7 було забезпечення портативності нечітких систем

керування. До появи цього стандарту виробники програмного забезпечення використовували власні пропрієтарні формати, що унеможливлювало перенесення логіки з одного середовища розробки в інше без повного переписування коду. FCL вирішує цю проблему, надаючи уніфікований текстовий синтаксис, зрозумілий як людині, так і машинним інтерпретаторам.

У розробленій системі використання FCL дозволяє відокремити логіку прийняття рішень (правила та функції належності) від програмного коду (Java Spring сервісів). Це означає, що для зміни параметрів пріоритезації (наприклад, зміни меж віку для категорії elderly або додавання нового правила) не потрібно перекомпілювати весь проєкт – достатньо лише відредагувати текстовий файл .fcl.

Структура FCL-файлу, що використовується у компоненті пріоритезації, складається з наступних обов'язкових блоків, які інкапсульовані в основний блок FUNCTION_BLOCK.

Блок оголошення змінних (VAR_INPUT та VAR_OUTPUT) визначає інтерфейс нечіткого контролера. Вхідними змінними у випадку розроблюваної технології є destruction (ступінь пошкодження), residents (кількість мешканців), age (вік), disability_group (група інвалідності) та is_large_family (багатодітність), а вихідною — priority (рівень пріоритету). Всі змінні типізуються як REAL.

Наступним є блок фазифікації (FUZZIFY), який описує процес перетворення чітких вхідних значень у нечіткі лінгвістичні змінні. Для кожної вхідної змінної задається діапазон (RANGE) та набір термів (TERM), які описуються функціями належності. У розробленій системі використано кусково-лінійне визначення функцій (пари точок X, Y), що дозволяє будувати трикутні та трапецієподібні форми.

Блок дефазифікації (DEFUZZIFY) визначає параметри вихідної змінної та метод перетворення нечіткого висновку назад у чітке число. У роботі використано метод центру ваги (METHOD : Centroid), який є найбільш поширеним у задачах пріоритезації, оскільки забезпечує плавний вихідний

сигнал. Також тут задається метод акумуляції (ACCU : MAX) та значення за замовчуванням (DEFAULT := 1) на випадок, якщо жодне правило не спрацює.

Останнім є блок правил (RULEBLOCK): Містить базу знань системи у вигляді продукційних правил «якщо-то» (IF ... THEN ...). Цей блок також визначає логічні оператори для машини виведення:

– AND:MIN(для об’єднання умов всередині правила використовується мінімум (перетин нечітких множин);

– OR:MAX (для об’єднання результатів різних правил використовується максимум (об’єднання множин);

– ACT:MIN (метод активації (імлікації), який обмежує вихідну функцію належності за рівнем істинності умови).

Загальний вміст конфігурації системи нечіткого виведення який збережено у FCL файлі наведено у лістингу 3.1.

Лістинг 3.1 – Вміст файлу

```
FUNCTION_BLOCK PriorityLogic
// Вхідні змінні
VAR_INPUT
  destruction : REAL;
  residents : REAL;
  age : REAL;
  disability_group : REAL;   // 0 – none, 1 – light, 2 – medium, 3 – severe
  is_large_family : REAL;    // 0 – no, 1 – yes
END_VAR
// Вихідна змінна
VAR_OUTPUT
  priority : REAL;
END_VAR

// (ступінь пошкодження житла)
FUZZIFY destruction
  RANGE := (1 .. 10);
  TERM low    := (1, 1) (2.5, 1) (4, 0);
  TERM medium := (3, 0) (4.5, 1) (5.5, 1) (7, 0);
  TERM high   := (6, 0) (7.5, 1) (10, 1);
END_FUZZIFY;

// (кількість мешканців)
FUZZIFY residents
  RANGE := (1 .. 10);
  TERM single := (1, 1) (1.5, 1) (2, 0);
```

```
  TERM normal := (1.5, 0) (2.5, 1) (4.5, 1) (5.5, 0);
  TERM many   := (4.5, 0) (6, 1) (10, 1);
END_FUZZIFY;

// age (вік власника)
FUZZIFY age
  RANGE := (18 .. 100);
  TERM young   := (18, 1) (18, 1) (28, 1) (35, 0);
  TERM middle  := (32, 0) (38, 1) (50, 1) (60, 0);
  TERM elderly := (58, 0) (70, 1) (100, 1) (100, 1);
END_FUZZIFY;

// disability_group (категоріальна змінна)
FUZZIFY disability_group
 RANGE := (0 .. 3);
  TERM none   := (0, 1) (0.5, 1) (1, 0);
  TERM light  := (0.5, 0) (1, 1) (1.5, 0);
  TERM medium := (1.5, 0) (2, 1) (2.5, 0);
  TERM severe := (2.5, 0) (3, 1) (3, 1);
END_FUZZIFY;

// is_large_family (багатодітна родина)
FUZZIFY is_large_family
  RANGE := (0 .. 1);
  TERM no  := (0, 1) (0.4, 1) (0.5, 0);
  TERM yes := (0.5, 0) (0.6, 1) (1, 1);
END_FUZZIFY;

DEFUZZIFY priority
  RANGE := (0 .. 10);
  TERM low    := (1, 1) (2, 1) (4, 0);
  TERM medium := (3, 0) (5, 1) (6, 1) (8, 0);
  TERM high   := (7, 0) (9, 1) (10, 1);
  METHOD : Centroid;
  ACCU : MAX;
  DEFAULT := 0;
END_DEFUZZIFY;

RULEBLOCK Main
  AND : MIN;
  OR  : MAX;
  ACT : MIN;

  RULE 1:  if destruction is high and residents is many and age is elderly
then priority is high;
  RULE 2:  if destruction is high and disability_group is severe then
priority is high;
  RULE 3:  if destruction is high and is_large_family is yes then priority
is high;
// *Далі правила продовжуються
END_RULEBLOCK
END_FUNCTION_BLOCK
```

4 ТЕСТУВАННЯ ТЕХНОЛОГІЇ ПРІОРИТЕЗАЦІЇ

4.1 Тестування методом «чорної скриньки»

Метою тестування є перевірка коректності роботи розробленої інформаційної технології пріоритезації виплат, а саме: валідація роботи алгоритму нечіткого виведення та інтеграції компонентів системи. Відповідно до стандарту ДСТУ 29119-1:2017 тестування проводиться методом «чорної скриньки» на рівні бізнес-логіки та методом перевірки інтерфейсу користувача [25].

4.1.1 Сценарій тестування

Для демонстрації роботи системи змоделюємо конкретний сценарій (Use Case), який відображає типову ситуацію для предметної області.

Сценарій тестування: До системи надійшла заявка на компенсацію від власника житла, яке зазнало руйнувань внаслідок бойових дій. Необхідно розрахувати пріоритет виплати на основі вхідних даних, що містяться в картці об'єкта та профілі власника.

Вхідні дані для тестового прикладу наведено у таблиці 4.1.

Таблиця 4.1 – Вхідні дані для тестового розрахунку

Змінна	Значення	Опис
destruction	8.5	Значні пошкодження (несучі стіни, дах)
residents	5	У житлі зареєстровано 5 осіб
age	72	Власник похилого віку
disability_group	2	II група інвалідності (середній ступінь)
is_large_family	1	Статус багатодітної сім'ї підтверджено

Розглянемо покроково, як компонент PriorityCalculator обробляє ці дані, використовуючи бібліотеку JFuzzyLite та налаштовані правила (файл .fcl), описані у розділі 4.

4.1.2. Фазифікація вхідних змінних

На цьому етапі чіткі числові значення перетворюються на лінгвістичні змінні з відповідними ступенями належності (μ).

Змінна destruction - згідно з функцією належності, значення 8.5 потрапляє у діапазон терму high:

$$\mu_{high}(8.5) = 1.0 \text{ (значення знаходиться на "плато" функції).}$$

$$\mu_{medium}(8.5) = 0.0.$$

Змінна residents: Згідно значення 5 є перехідним між normal та many.

Для терму many (трапеція: 4.5, 6, 10, 10):

$$\mu_{many}(5) = \frac{5-4.5}{6-4.5} \approx 0.33.$$

Для терму normal:

$$\mu_{normal}(5) = \frac{5.5-5}{5.5-4.5} \approx 0.67.$$

Змінна age - згідно з рис. 3.3, значення 72 потрапляє у діапазон терму elderly:

$$\mu_{elderly}(72) = 1.0 \text{ (оскільки } 72>70).$$

Змінна disability_group:

$$\mu_{medium}(2) = 1.0 \text{ (пік трикутної функції).}$$

Змінна is_large_family:

$$\mu_{yes}(1) = 1.0.$$

4.1.3 Агрегація та спрацювання правил

Система застосовує базу знань, що складається з 31 правила. Оскільки для вхідних даних деякі змінні мають ступінь належності > 0 до кількох термів (наприклад, residents), активується значна частина правил.

Для агрегації використовується оператор MIN (для логічного «І» у посилці правила) та MAX (для акумуляції впливу правил на вихідну змінну). Ключові правила, що активуються та визначають остаточний пріоритет наведено у таблиці 5.2.

Таблиця 4.2 – Правила що активуються для вхідних даних

№ Правила	Формула правила	Рівень активації (α)	Вихідний терм
RULE 23	if age is elderly and destruction is high then priority is high	min(1.0, 1.0) = 1.0	High
RULE 9	if destruction is high and disability_group is medium then priority is high	min(1.0, 1.0) = 1.0	High
RULE 11	if is_large_family is yes and disability_group is medium then priority is high	min(1.0, 1.0) = 1.0	High
RULE 29	if destruction is high then priority is high	1.0 = 1.0	High
RULE 31	if is_large_family is yes then priority is high	1.0 = 1.0	High
RULE 30	if disability_group is medium then priority is medium	1.0 = 1.0	Medium
RULE 26	if residents is many and destruction is high then priority is high	min(0.33, 1.0) = 0.33	High
RULE 10	if residents is many and disability_group is medium then priority is high	min(0.33, 1.0) = 0.33	High
RULE 22	if is_large_family is yes and residents is many then priority is high	min(1.0, 0.33) = 0.33	High

У результаті акумуляції (методом MAX) отримано, що, ступінь належності до терму High:

$$\mu_{High} = \max(1.0,1.0,1.0,1.0,1.0,0.33) = 1.0,$$

а ступінь належності до терму Medium:

$$\mu_{Medium} = 1.0 \text{ (від RULE 30)}$$

Отримана результуюча нечітка множина для вихідної змінної priority максимально насичена в термах High та Medium, що логічно відображає наявність кількох критичних факторів у заявці.

4.1.4 Дефазифікація

Для отримання чіткого числового значення пріоритету використовується метод Центру Ваги, оскільки він забезпечує найбільш плавний і справедливий вихід. Поєднання максимальної активації термів High та Medium зі зміщенням впливу High дає наступний розрахунковий результат:

$$\text{Priority} \approx 8.9$$

4.2 Аналіз результатів роботи програмного забезпечення

Для перевірки інтеграції було виконано запит до API розробленого компоненту. На рисунку 4.1 зображено тіло HTTP-запиту (JSON), що надсилається основною системою до контролера.

У відповідь сервер повертає об'єкт Payment, що містить розрахований пріоритет.

```json
{
  "compensationSum": 1200000.0,
  "dwellingOwner": {
    "id": 1001,
    "firstName": "Олена",
    "lastName": "Сіренька",
    "middleName": "Михайлівна",
    "phoneNumber": "+380501234567",
    "email": "ivan.kovalenko@example.com",
    "IBAN": "UA123456000000000000000000001",
    "birthDate": "1943-05-15",
    "disabilityGroup": 2
  },
  "dwelling": {
    "id": 2001,
    "isLargeFamily": false,
    "residentsNumber": 5,
    "destructionLevel": 8.5,
    "address": {
      "regionName": "Запорізька область",
      "localityName": "с. Долинське",
      "streetName": "вул. Церковна",
      "buildingNumber": "10",
      "apartmentNumber": null
    }
  }
}
```


Рисунок 4.1 – Дані про власника житла та житло

Логи, на яких зображено увесь цикл від отримання вхідних даних до збереження готової пріоритетної виплати наведено на рис. 4.2.

Рисунок 4.2 – Логи сервісу

На рисунку 4.3 представлено головне вікно веб-інтерфейсу, яке є робочим місцем бухгалтера. Інтерфейс розроблено з використанням технології Thymeleaf для відображення даних, отриманих від REST API компонента системи.

У верхній частині сторінки розміщено панель інструментів, що дозволяє фільтрувати заявки за географічними ознаками («Область», «Населений пункт») та статусом виплати («Статус»). Це прямо відповідає вимозі D-1.1.2, згідно з якою система повинна надавати можливість переглядати відфільтровані списки.

Основна область екрану містить список карток заявок, відсортованих за спаданням пріоритету. Таке сортування підтверджує коректність роботи алгоритму ранжування: заявки з критичними показниками знаходяться у верхній частині списку.

Кожна картка містить вичерпний набір даних, необхідних для прийняття рішення, що реалізує вимогу D-1.1.1:

– фінансові дані: сума виплати та поточний статус;

– показник пріоритету, який є результатом роботи нечіткої логіки;

– деталі про житло (адреса та рівень пошкодження житла);

– персональна інформація, така як: ПІБ власника, вік, група інвалідності, склад сім'ї.

Наявність кнопки «Оформити виплату коштів» дозволяє бухгалтеру ініціювати транзакцію безпосередньо зі списку.

Червона кнопка PDF дозволяє експортувати отримані результати у PDF, що відповідає заявленим вимогам.

Як видно з результатів, заявка отримала пріоритет 8.9, що поміщає її у верхню частину списку виплат.

4.3 Навантажувальне тестування технології

Тестування продуктивності ІС реалізовано за допомогою Apache JMeter – open-source рішення для навантажувального аналізу. Основна функція ПЗ

полягає у генерації запитів до сервера, що імітують активність реальних користувачів, для визначення меж стабільності системи [26].

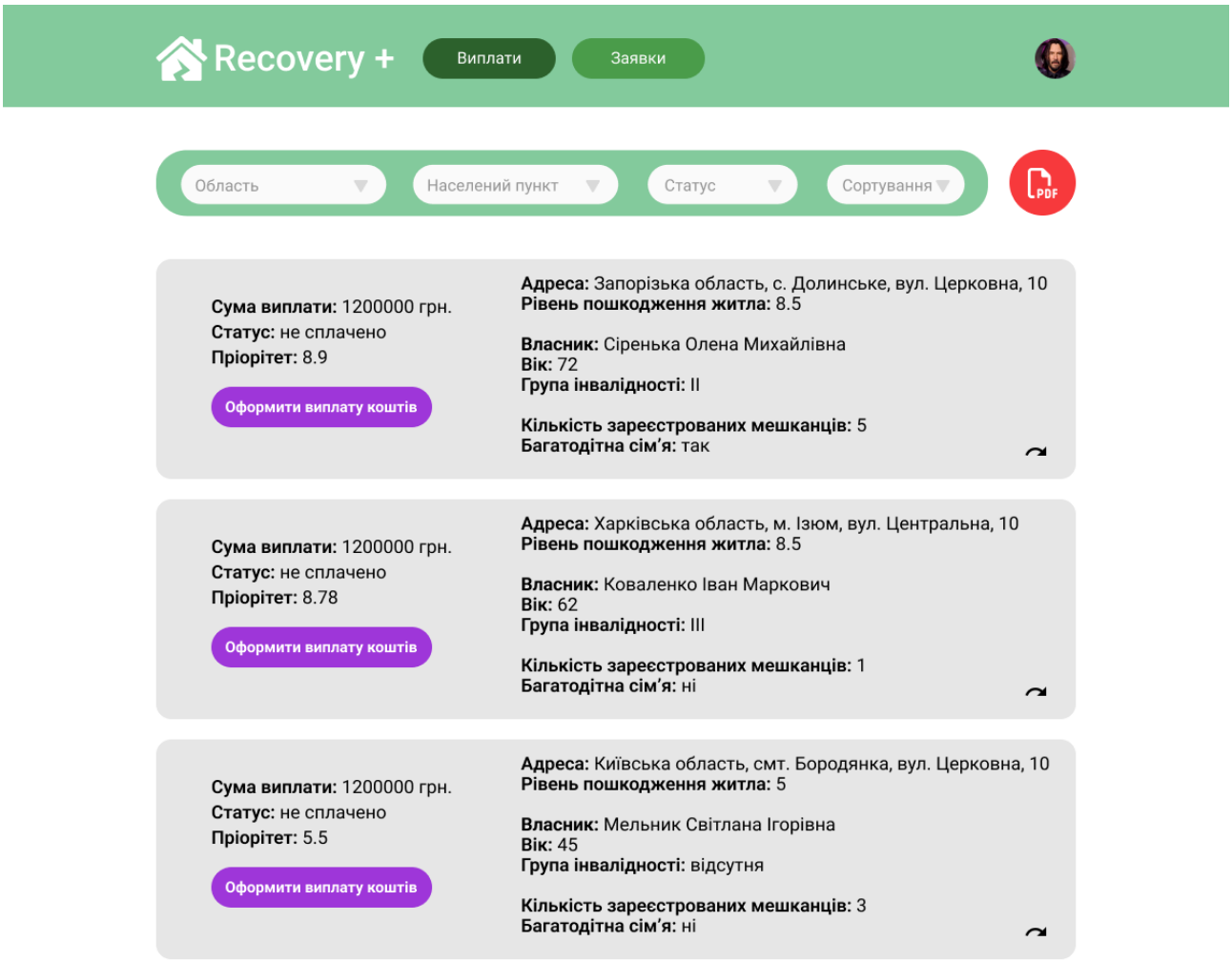

Рисунок 4.3 – Список виплат

Серед ключових переваг інструменту: підтримка різноманітних протоколів (від HTTP до JDBC) та можливість створення складних алгоритмів тестування у графічному середовищі.

Для перевірки навантаження обрано запит, який найчастіше буде використовуватися при роботі системи, а саме: обчислення пріорітету та збереження інформації у БД та відображення списку виплат за населеним пунктом та статусом. Відображення цих запитів у форматі MongoDB Query Language (MQL) наведено у лістингу 4.1.

Лістинг 4.1 – Навантажені запити до БД

```
db.payment.insertOne({})
db.payment.find({ "dwelling.address.localityName": "м. Ізюм", "statuses.statusName": "NOT_PAID" })
```

Далі у программі Jmeter створено наступну конфігурацію, яка визначає групу потоків(користувачів), яку зображено на рис. 4.4. Для тестування обрано 2000 користувачів які будуть одночасно звертатися за адресою сервісу http://localhost:8081/api/payment ви кикликати POST та GET методи. Конфігурація HTTP запиту у середовищі Jmeter наведено на рис. 5.3.

Для 2000 запитів отриманих за 10 секунд отримано графік, який наведено на рис 4.4. Пропускна здатність (Throughput) 5 359,536 запитів у хвилину, що є непоганим показником для розробленої системи. Однак слід вважати, що результати тестування системи на локальному сервері та у реальних умовах, можуть суттєво відрізнятися.

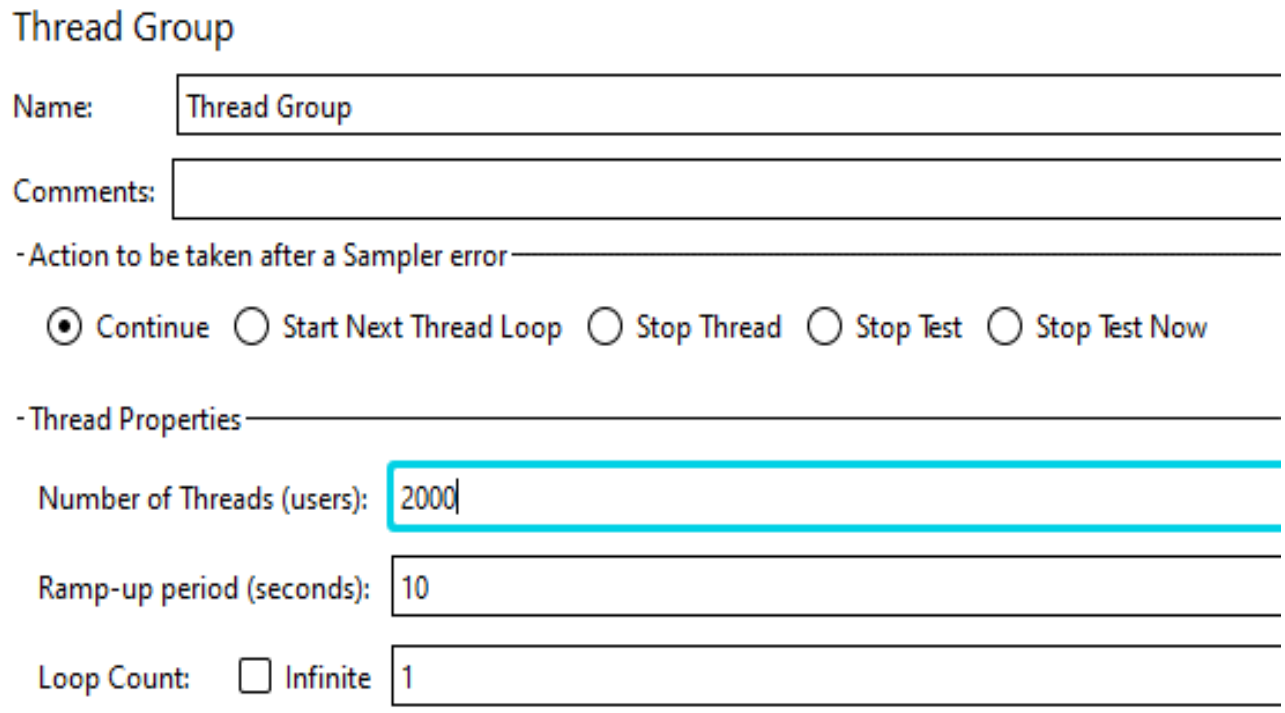

Рисунок 4.4 – Конфігурація навантаження

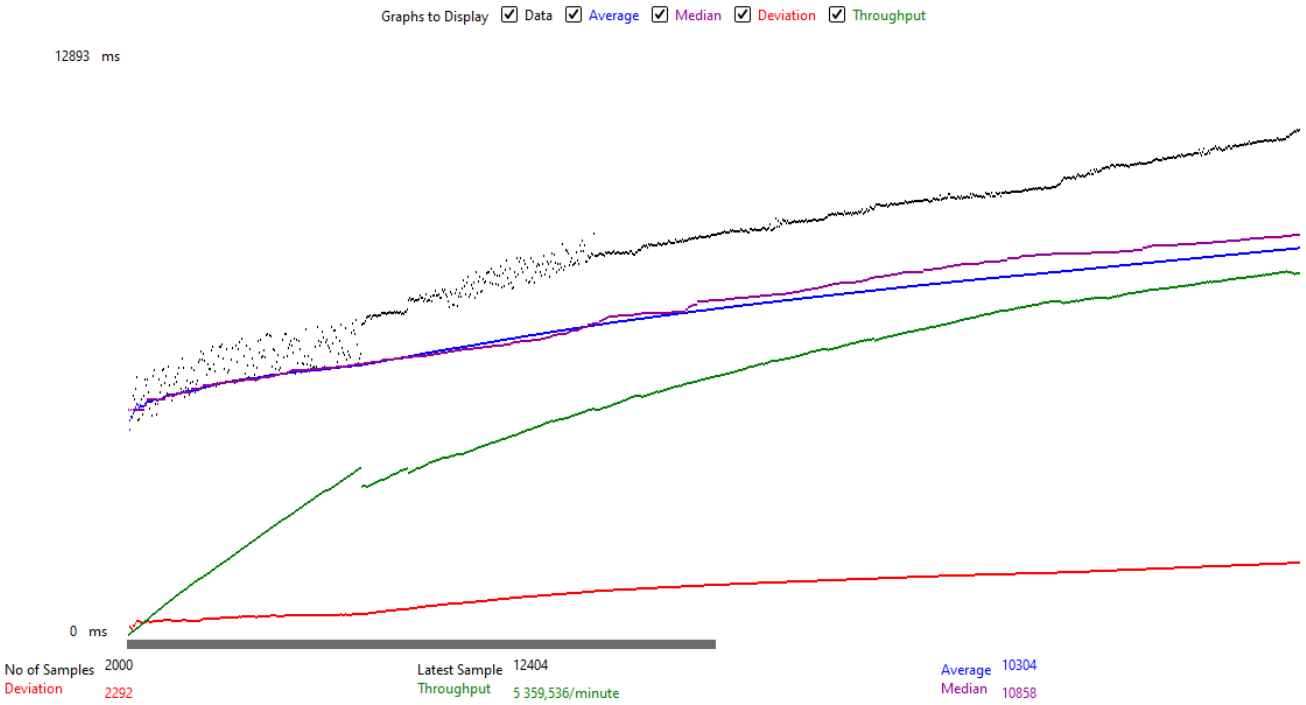

Рисунок 4.4 – Графік навантажень

4.4 Аналіз обмежень поточної реалізації

Розроблена інформаційна технологія базується на системі нечіткого виведення типу Мамдані. Як було зазначено раніше, цей підхід є оптимальним для задач із високим рівнем невизначеності та відсутністю накопиченої статистики. Однак, з точки зору теорії системного аналізу та довгострокової експлуатації, поточне рішення має низку теоретичних недоліків, які визначають вектор подальшого вдосконалення системи.

Перша проблема – статичність бази знань. Описана у форматі FCL логіка є фіксованою. Система не здатна самостійно адаптуватися до змін у зовнішньому середовищі (наприклад, якщо зміниться політика фонду щодо пріоритетів або різко зросте кількість заявок з певної категорії).

Наступною проблемою є curse of dimensionality (прокляття розмірності). При збільшенні кількості вхідних критеріїв кількість правил зростає експоненціально. Поточна модель використовує 5 змінних і близько 30 правил, але додавання нових факторів (наприклад, доходу сім'ї чи наявності іншого житла) значно ускладнить адміністрування бази знань.

Для вирішення проблеми масштабованості та підвищення точності пропонується впровадити дворівневу архітектуру, залучивши методи машинного навчання, розглянуті в аналітичному огляді.

Рівень 1 – Machine Learning. Використання алгоритмів градієнтного бустингу (XGBoost або LightGBM) для виявлення аномалій та попередньої сегментації заявок. ML-модель може аналізувати історичні дані для виявлення патернів шахрайства або нетипових значень у полях «рівень пошкодження».

Рівень 2 – Fuzzy Logic. Відфільтровані та верифіковані заявки передаються на вхід нечіткому контролеру для остаточного розрахунку пріоритету.

Такий підхід дозволить використовувати сильні сторони ML (робота з великими даними, виявлення прихованих залежностей) та FCL (прозорість та соціальна орієнтованість).

Хоча поточна архітектура на базі MongoDB та JFuzzyLite демонструє високу швидкодію (15 мс на запит), при масштабуванні на глобальний рівень з мільйонами заявок, доцільно розглянути перехід від моделі Мамдані до моделі Сугено.

Модель Сугено замінює нечіткі множини у висновках правил на лінійні функції. Це значно спрощує процес дефазифікації, роблячи його обчислювально ефективнішим, що є критичним для систем реального часу з високим навантаженням.

ВИСНОВКИ

У ході виконання роботи проведено комплексне дослідження проблеми визначення пріоритетності виплат у системі обліку пошкодженого житла, що є актуальною задачею соціально-гуманітарного спрямування в умовах воєнних дій та обмежених ресурсів. У рамках роботи було виконано аналіз предметної області, визначено ключові соціальні та технічні параметри, що впливають на черговість виплати компенсацій, а також досліджено сучасні математичні, алгоритмічні та інтелектуальні методи оцінювання пріоритетів.

Проведений огляд наукових джерел дав змогу визначити, що класичні формалізовані методи, такі як лінійна модель або алгоритми пріоритетів, мають обмежені можливості при роботі з якісними, нечіткими або суб’єктивними даними. Методи машинного навчання забезпечують високу точність прогнозування, проте потребують великого обсягу навчальних вибірок та не завжди забезпечують інтерпретованість рішень. Найбільш придатним підходом для задачі пріоритезації компенсацій виявилася нечітка логіка, яка дозволяє працювати з лінгвістичними оцінками, експертними знаннями та невизначеністю у вихідних даних.

У результаті дослідження було розроблено ІТ пріоритезації виплат, що інтегрується в інформаційну систему обліку пошкодженого житла. Для цього було побудовано систему нечіткого логічного виведення типу Мамдані, що включає п’ять основних вхідних змінних: ступінь пошкодження житла, кількість зареєстрованих мешканців, вік власника, групу інвалідності та ознаку багатодітності. Для кожної змінної визначено відповідні функції належності трикутного та трапецевидного типу.

За допомогою фунціонального та UML моделювання змодельовано інформаційну технологію, визначено архітектуру та побудовано базу нечітких правил, яка відображає експертну логіку прийняття рішень у гуманітарних системах та дозволяє формувати інтегральний показник пріоритетності виплати.

Використовуючи основні методи тестування, такі як метод «чорної скриньки» та навантажувальне, вдалося з’ясувати, що технологія як компонент системи обліку пошкодженого житла має точність, ідентичну теоретичній та здатна витримувати велике навантаження (з урахуванням запуску у локальній мережі).

Проведено аналіз перспектив розвитку технології де визначено, що використання машинного навчання та переходу з системи нечіткого виведення від Мамдамі до Сугено значно підвищить її стабільність та швидкодію.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ребров В. С. Методи пріоритезації виплат в системі обліку пошкодженого житла / Розвиток сучасної науки: актуальні питання теорії та практики: матеріали IX Всеукраїнської студентської наукової конференції, м. Київ, 21 листопада, 2025 рік / ГО «Молодіжна наукова ліга». Вінниця: ТОВ «УКРЛОГОС Груп», 2025. С. 509-510.

2. Постанова Кабінету Міністрів України про затвердження порядку надання компенсації за знищені об’єкти нерухомого майна від 30.05.2022. URL: https://zakon.rada.gov.ua/laws/show/600-2023-п (дата звернення: 02.11.2025).

3. Ismail N., Ahmad H. Consumer debt management among government employees: cognitive, emotional and behavioural. e-Academia Journal. 2023. Vol. 12, no. 2. P. 212–223. URL: https://doi.org/10.24191/e-aj.v12i2.23992 (дата звернення 05.11.2025).

4. Analytic Hierarchy Process. Multiple Attribute Decision Making. 2011. P. 29–42. URL: https://doi.org/10.1201/b11032-6 (дата звернення: 05.11.2025).

5. International Conference on Fuzzy Sets Theory and its Applications (4th 1998 Liptovský Ján, Slovakia). Fuzzy sets: Theory and applications / ed. by K. Anna, K. Martin. Bratislava : Mathematical Institute, Slovak Academy of Sciences, 1999. 382 p.

6. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2016. P. 785–794

7. Hajihosseinlou M. A Novel Scheme for Mapping of MVT-Type Prospectivity: LightGBM, a Highly Efficient Gradient Boosting Machine Learning Algorithm. Natural Resources Research. 2023. URL: https://doi.org/10.1007/s11053-023-10249-6 (дата звернення 06.11.2025).

8. Molnar C. Interpretable Machine Learning. lulu.com, 2020. 318 p.

9. Decision Making: Descriptive, Normative, and Prescriptive Interactions / S. French et al. The Journal of the Operational Research Society. 1990. Vol. 41, no. 3. P. 263. URL: https://doi.org/10.2307/2583822 (дата звернення: 07.11.2025).

10. Davis, R. I. et al. A review of priority assignment in real-time systems // Journal of Systems Architecture. – 2016. – Vol. 65. – P. 64–82. – URL: https://doi.org/10.1016/j.sysarc.2016.04.002 (дата звернення: 12.11.2025).

11. Jacobson, E. U., Argon, N. T., Ziya, S. Priority Assignment in Emergency Response // Operations Research. – 2012. – Vol. 60, No. 1. – P. 119–131. – URL: https://doi.org/10.1287/opre.1120.1075 (дата звернення: 14.11.2025).

12. Shen, Z., Wang, Z., Zhu, X., Fain, B., Munagala, K. Fairness in the Assignment Problem with Uncertain Priorities // arXiv preprint. – 2023. – URL: https://doi.org/10.48550/arXiv.2301.13804 (дата звернення: 14.11.2025).

13. Chala, O., Bilova, T., Pobizhenko, I., Ostapenko, O. Fuzzy ontological model of knowledge representation for the humanitarian response // CEUR Workshop Proceedings (COLINS). – 2025. – URL: https://doi.org/10.31110/COLINS/2025-3/004 (дата звернення: 15.11.2025).

14. Ma'arif, A., Wijaya, S. A., Murni, M., Suwarno, I. Analysis and Performance Comparison of Fuzzy Inference Systems in Handling Uncertainty: A Review // Journal of Research and Computing. – 2023. – Vol. 5, No. 4. – URL: https://doi.org/10.18196/jrc.v5i4.22123 (дата звернення: 2025-11-15).

15. Gougas V. Issues in Process Modelling: IDEF, UML and EPC Models. Akakia Publications, 2022.

16. ISO 29110. URL: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:29110:-5-4 (дата звернення: 16.11.2025).

17. Sundaramoorthy S. Uml Diagramming. Taylor & Francis Group, 2022.

18. Wolff E. Microservices: Flexible Software Architecture. Pearson Education, Limited, 2021.

19. Spring Framework Documentation :: Spring Framework. URL: https://docs.spring.io/spring-framework/ (дата звернення: 17.11.2025).

20. FuzzyLite: The FuzzyLite Libraries for Fuzzy Logic Control https://fuzzylite.com/documentation (дата звернення: 17.11.2025).

21. MongoDB Documentation. URL: https://www.mongodb.com/docs (дата звернення: 17.11.2025).

22. Tutorial: Using Thymeleaf. URL: https://www.thymeleaf.org/documentation.html (дата звернення: 17.11.2025).

23. Hackolade Documentation. URL: https://hackolade.com/help/Tutorial.html (дата звернення: 17.11.2025).

24. IEC 61131-7: Fuzzy Control Programming. URL: www.plcopen.org/standards/logic/iec-61131-7 (дата звернення: 18.11.2025).

25. ДСТУ ISO/IEC/IEEE 29119-1:2017. Інженерія систем і програмних засобів. Тестування програмних засобів. Частина 1. Поняття та визначення (ISO/IEC/IEEE 29119-1:2013, IDT). – Київ : ДП «УкрНДНЦ», 2017. – 47 с.

26. Apache JMeter - User's Manual. URL: https://jmeter.apache.org/usermanual/index.html (дата звернення: 30.11.2025).