МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Іванчук Нікіта Анатолійович
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.9:336.7

КВАЛІФІКАЦІЙНА РОБОТА

Проектування та розробка інформаційної системи моніторингу санкційних ризиків у банківських операціях
(тема роботи)

122 «Комп'ютерні науки»
(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____
(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Гіваргізов Інвія Геннадійович
(прізвище, ім'я, по батькові)
к.е.н., старший викладач кафедри
(науковий ступінь, вчене звання)

Житомир – 2026

Висновок кафедри комп'ютерних технологій і моделювання систем
за результатами попереднього захисту: допущений

Протокол засідання кафедри комп'ютерних технологій і моделювання систем
№ 20 від «05» червня 2026 р.

Завідувач кафедри комп'ютерних технологій і моделювання систем
к. пед н., доцент _____ КОВАЛЬЧУК М. О.
(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)
«05» червня 2026

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище ,ім'я, по батькові)
кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____
за шкалою ECTS _____
за національною шкалою _____

Секретар ЕК
лаборант кафедри _____ КОРОЛЬЧУК В. В.
(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)

АНОТАЦІЯ

Іванчук Н. А. Проектування та розробка інформаційної системи моніторингу санкційних ризиків у банківських операціях. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 122 – комп'ютерні науки. – Поліський національний університет, Житомир, 2026.

Кваліфікаційна робота присвячена вирішенню актуальної науково-практичної проблеми – автоматизації процесів фінансового моніторингу та підвищенню точності ідентифікації санкційних ризиків в умовах строгих вимог до збереження банківської таємниці. У роботі досліджено обмеження існуючих хмарних інформаційно-аналітичних сервісів та обґрунтовано необхідність створення автономного локального програмного рішення.

Науковим ядром роботи є дослідження та застосування методів обробки текстової інформації на основі математичної моделі нечіткої логіки. Досліджено проблему ідентифікації підсанкційних осіб за умови наявності у платіжних документах друкарських помилок, зміненого порядку слів та різної транслітерації. Для вирішення цього завдання реалізовано алгоритм на базі метрики подібності рядків. У ході дослідження експериментально визначено та обґрунтовано оптимальне порогове значення чутливості алгоритму на рівні 85%, що забезпечує необхідний баланс між гарантованим виявленням ризикових транзакцій та мінімізацією хибних спрацьовувань. Додатково досліджено та інтегровано механізм багаторівневої перевірки, що включає семантичний аналіз стоп-слів у призначенні платежу.

Ключові слова: банківські операції, інформаційна система, санкції, фінансовий моніторинг, нечітка логіка.

SUMMARY

Ivanchuk N. A. Design and development of an information system for monitoring sanctions risks in banking operations. – Qualification paper as a manuscript.

Qualification paper for the educational degree of Bachelor in specialty 122 – Computer Science. – Polissia National University, Zhytomyr, 2026.

The qualification paper is devoted to solving a topical scientific and practical problem – the automation of financial monitoring processes and improving the accuracy of identifying sanctions risks under strict requirements for maintaining banking secrecy. The paper investigates the limitations of existing cloud-based information and analytical services and justifies the need to create an autonomous local software solution.

The scientific core of the work is the research and application of text information processing methods based on a fuzzy logic mathematical model. The problem of identifying sanctioned individuals under the conditions of typographical errors, altered word order, and different transliteration in payment documents is investigated. To solve this task, an algorithm based on a string similarity metric has been implemented. During the research, the optimal sensitivity threshold of the algorithm at the level of 85% was experimentally determined and justified, which provides the necessary balance between the guaranteed detection of risky transactions and the minimization of false positives. Additionally, a multi-level verification mechanism was investigated and integrated, which includes a semantic analysis of stop words in the payment details.

Keywords: banking operations, information system, sanctions, financial monitoring, fuzzy logic.

ЗМІСТ

ВСТУП

Кваліфікаційна робота: 40 с., 23 рис., 1 табл., 11 джерел, 2 додатки

Актуальність теми. В умовах стрімкої цифровізації банківського сектору та посилення міжнародного й національного санкційного тиску, питання безперервного та ефективного фінансового моніторингу стає критично важливим. Забезпечення нормативної відповідності вимагає від підрозділів комплаєнсу миттєвої ідентифікації ризикових операцій. Використання популярних зовнішніх онлайн-сервісів для потокової перевірки кожного платежу часто несе загрозу розголошення банківської таємниці. Тому розробка повністю автономної, локальної автоматизованої інформаційної системи, здатної безпечно обробляти транзакції на предмет санкційних ризиків із врахуванням можливих друкарських помилок у документах, є надзвичайно актуальним завданням.

Мета і завдання роботи. Метою роботи є проектування та розробка локальної інформаційної системи моніторингу санкційних ризиків у фінансових операціях із застосуванням методів нечіткого пошуку.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати предметну область, інформаційні потреби підрозділів фінансового моніторингу та обґрунтувати необхідність локальної обробки даних;

- спроектувати функціональну архітектуру системи (UML, IDEF0) та інфологічну структуру реляційної бази даних;

- реалізувати математичну модель та програмні алгоритми на базі нечіткої логіки для ідентифікації збігів (включаючи аналіз призначення платежу);

- створити зручний корпоративний графічний інтерфейс для користувачів застосунку;

- провести експериментальне тестування системи на створених вибірках транзакцій, включаючи сценарії хибних спрацьовувань.

Об’єкт дослідження: процес автоматизованого моніторингу та ідентифікації санкційних ризиків у фінансових операціях.

Предмет дослідження: методи, алгоритми та програмні засоби проектування автоматизованої системи контролю санкційних ризиків.

Методи дослідження.

1. **Методи системного аналізу та проектування (IDEF0, UML)** - використано для розробки архітектури програмного комплексу, формалізації бізнес-процесів та розмежування ролей користувачів.

2. **Теорія реляційних баз даних** - застосовано для побудови логічної структури сховища даних на основі SQLite та забезпечення цілісності інформації журналу аудиту.

3. **Методи нечіткої логіки та метрики подібності рядків** - використано для реалізації алгоритму пошуку санкційних збігів, що забезпечує ідентифікацію непрямих збігів у реквізитах із заданим порогом чутливості (85%).

4. **Методи програмної інженерії та об'єктно-орієнтованого програмування** - застосовано під час написання коду застосунку (Python) та розробки графічного інтерфейсу (PyQt6).

5. **Алгоритмічний парсинг** - використано для автоматизованого зчитування та імпорту нормативних даних із CSV-файлів (санкційних списків РНБО) до внутрішньої бази.

Основні положення роботи та результати досліджень висвітлено у матеріалах Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених «Інформаційні технології та моделювання систем» (м. Житомир, 22 травня 2026 року) та XVIII Міжнародної науково-практичної конференції «Концептуальні шляхи розвитку науки та освіти» (м. Львів, 29–30 травня 2026 року).

Практичне значення отриманих результатів. Створена інформаційна система є повністю готовим до впровадження програмним продуктом. Вона дозволяє суттєво автоматизувати процес потокової перевірки транзакцій, гарантуючи при цьому стовідсоткову конфіденційність даних. Впровадження системи мінімізує людський фактор під час виявлення непрямих збігів із санкційними списками РНБО, пришвидшує роботу операторів завдяки ергономічному інтерфейсу та забезпечує жорсткий, прозорий облік усіх прийнятих рішень через вбудований журнал аудиту.

.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ МОНІТОРИНГУ САНКЦІЙНИХ РИЗИКІВ

1.1 Аналіз інформаційних потреб і визначення предметної області

Сучасний банківський сектор функціонує в умовах жорсткого нормативного регулювання та постійно зростаючих вимог до фінансового моніторингу. Особливої актуальності набуває питання безумовного дотримання санкційного законодавства, що покладає на підрозділи внутрішнього контролю завдання безперервного аналізу фінансових потоків. Предметна область даного дослідження охоплює процеси автоматизованої обробки банківських даних з метою ідентифікації ризик-факторів, пов’язаних із транзакціями підсанкційних осіб.

Для ефективного функціонування та виконання нормативних вимог, інформаційна система підрозділу внутрішнього контролю повинна задовольняти такі базові інформаційні потреби:

1. **Вхідні дані транзакцій:** система повинна приймати та структурувати інформацію з платіжних документів (код ЄДРПОУ або ІПН, найменування платника та отримувача, призначення платежу, сума операції тощо).

2. Нормативно-довідкова інформація: система потребує інтеграції з актуальною базою даних санкційних списків (зокрема, реєстрами РНБО України).

3. **Аналітичне ядро:** система повинна забезпечувати високошвидкісне порівняння реквізитів транзакцій із даними санкційних списків. Оскільки в платіжних документах часто трапляються друкарські помилки або різне написання ПІБ, система потребує використання моделей нечіткої логіки для виявлення часткових збігів, а не лише точного текстового пошуку.

Для визначення переваг та недоліків сучасних засобів моніторингу, що використовуються для аналізу відкритих реєстрів та баз даних, було розглянуто провідні інформаційно-аналітичні платформи.

1. YouControl [1] – це аналітична онлайн-система перевірки контрагентів, яка бере дані з понад 180 відкритих реєстрів. У контексті фінансового моніторингу ключовим є модуль «YouControl Compliance». Платформа володіє найповнішою

базою зв’язків, що дозволяє виявляти не лише пряме входження до санкційних списків, а й приховані зв'язки. Проте, ця система орієнтована на глибокий аналіз. Для потокової перевірки кожного окремого платежу її використання може бути економічно недоцільним через високу вартість API-запитів та надмірність отримуваної інформації, тоді як оператору потрібен лише результат (є збіг / немає збігу). На рисунку 1.1 наведено головну сторінку онлайн-сервісу YouControl.

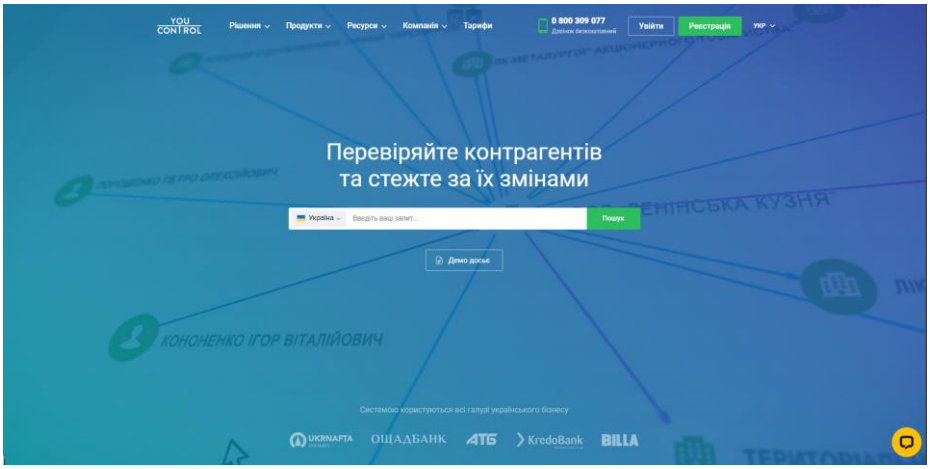

Рис. 1.1 – Онлайн сервіс YouControl

2. Vkursi [2] – сервіс бізнес-розвідки, який надає розширене досьє компаній, інформацію про судові рішення та санкційні обмеження. Платформа ефективно структурує дані про юридичних осіб та має інструменти для масової перевірки кодів ЄДРПОУ. Однак, як і YouControl, це зовнішній хмарний сервіс. Для банківської установи передача чутливих даних про транзакції (суми, призначення платежу, рахунки) на зовнішні сервери несе суттєві ризики порушення банківської таємниці та інформаційної безпеки. На рисунку 1.2 зображено голову сторінку онлайн-сервісу Vkursi.

Рис. 1.2 – Платформа Vkursi

3. Opendatabot [3] – сервіс моніторингу реєстраційних даних українських компаній. Це найбільш масовий та простий у використанні інструмент, який забезпечує швидкий пошук за кодом ЄДРПОУ. Недоліком є те, що його функціонал переважно орієнтований на взаємодію через онлайн платформи та веб-інтерфейс для малого та середнього бізнесу. Для банківського моніторингу, де потрібна обробка транзакцій з урахуванням орфографічних похибок, його базових алгоритмів недостатньо. На рисунку 1.3 зображено головний екран сервісу Opendatabot.

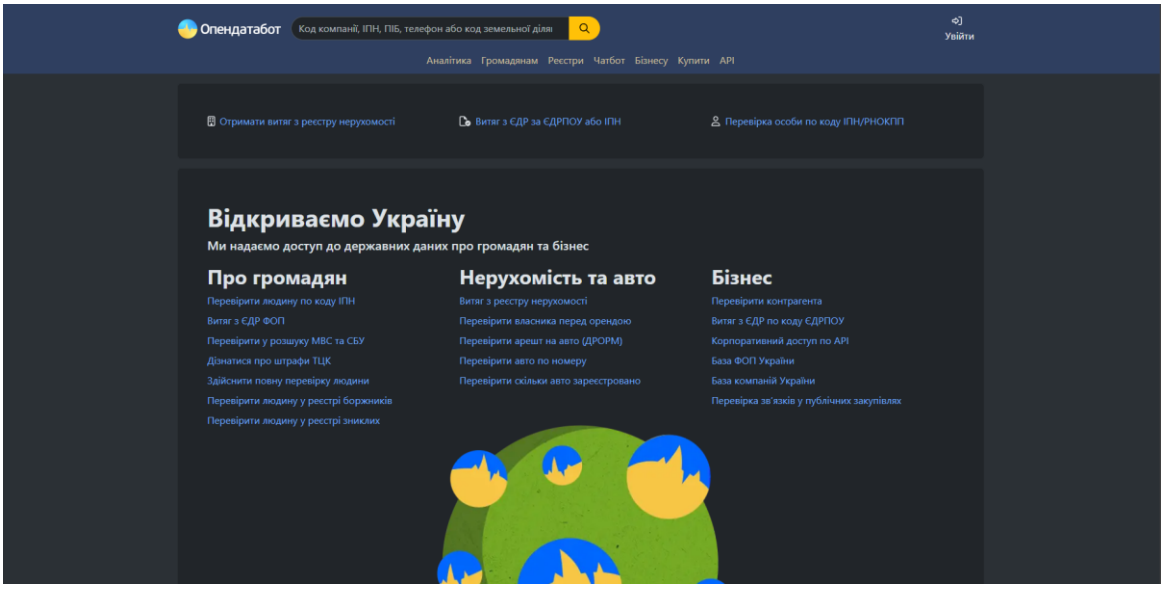

Рис. 1.3 – Головний екран сервісу Opendatabot

Для узагальнення результатів аналізу та наочного представлення переваг і недоліків існуючих платформ у контексті задач фінансового моніторингу, сформовано порівняльну таблицю. Вона дозволяє обґрунтувати доцільність розробки власного локального рішення.

Таблиця 1.1 – Порівняльний аналіз інформаційно-аналітичних платформ

Назва сервісу	**Переваги**	**Недоліки**
YouControl	1. Агрегація даних з понад 180 відкритих реєстрів. 2. Найповніша база зв'язків для виявлення прихованих даних. 3. Глибокий аналіз контрагентів.	1. Орієнтація на глибинний аналіз, що створює надмірність інформації для швидкої перевірки. 2. Економічна недоцільність для потокової обробки кожного платежу через високу вартість API-запитів. 3. Ризики порушення банківської таємниці при передачі даних на зовнішні сервери.
Vkursi	1. Надання розширеного досьє компаній та інформації про судові рішення/санкції. 2. Ефективне структурування даних про юридичних осіб. 3. Наявність інструментів для масової перевірки кодів ЄДРПОУ.	1. Є зовнішнім хмарним сервісом, що вимагає передачі чутливих даних (суми, призначення платежу, рахунки) поза межі закритого контуру банку. 2. Суттєві ризики для інформаційної безпеки установи.
Opendatabot	1. Максимальна простота у використанні. 2. Забезпечення швидкого пошуку за ключовим ідентифікатором (кодом ЄДРПОУ). 3. Масовість та доступність інструменту.	1. Орієнтація функціоналу переважно на малий та середній бізнес (робота через онлайн-платформи та веб-інтерфейс). 2. Недостатність базових алгоритмів пошуку для обробки транзакцій із можливими орфографічними похибками чи зміненою транслітерацією.

Проведений аналіз показує, що сучасні інструменти обробки відкритих даних орієнтовані на різні категорії користувачів і мають як сильні сторони, так і обмеження. Використовувати їх для постійної перевірки кожної банківської транзакції буде складно, а передача даних на зовнішні сервери несе ризики для інформаційної безпеки банку. Врахування цих особливостей є необхідним етапом під час проектування власної локальної системи моніторингу санкційних ризиків у банківських операціях, яка працюватиме на основі моделей нечіткої логіки.

1.2 Визначення функціональних вимог до інформаційної системи

На основі аналізу інформаційних потреб підрозділів внутрішнього контролю банку було сформовано перелік чітких вимог до можливостей та функцій розроблюваного програмного забезпечення. Ці вимоги визначають, що саме повинна робити система для забезпечення надійного контролю санкційних ризиків.

Основними функціональними вимогами до системи є:

1. Керування санкційними даними: система повинна мати функціонал для завантаження та оновлення актуальних санкційних списків РНБО України з файлів формату CSV. Отримані дані мають автоматично структуруватися та зберігатися у локальній базі даних.

2. Алгоритм перевірки: основна логіка перевірки платіжних документів має базуватися на використанні нечіткої логіки. Це необхідно для порівняння ПІБ або назв організацій з транзакціями із санкційною базою даних. Алгоритм повинен знаходити часткові збіги за заданим порогом чутливості (85%), що дозволить успішно виявляти ризики навіть за наявності друкарських помилок або різної транслітерації в документах.

3. Зручний та зрозумілий інтерфейс користувача: графічна оболонка застосунку повинна бути зручною та зрозумілою для оператора, забезпечувати швидкий доступ до основних модулів системи та не перевантажувати користувача зайвою інформацією.

4. Журнал аудиту: інформаційна система повинна автоматично реєструвати всі дії оператора (запуск перевірок, зміну статусів операцій) у захищеному журналі для забезпечення повної прозорості контрольних заходів.

5. Доступ до бази санкційних списків: система повинна надавати оператору зручний інтерфейс для прямого перегляду завантаженої нормативно-довідкової інформації (реєстрів РНБО). Це необхідно для того, щоб у разі виникнення нестандартних або спірних ситуацій під час автоматичного моніторингу, фахівець мав можливість здійснити ручну верифікацію даних та самостійно перевірити реквізити конкретної особи чи організації безпосередньо у санкційному списку.

Окрім функціональних можливостей, до системи висуваються важливі технічні вимоги:

- **Автономність та конфіденційність:** оскільки система працює з банківською таємницею, всі обчислення та збереження даних мають відбуватися суворо локально без передачі інформації на будь-які сторонні сервери чи хмарні сервіси через мережу Інтернет.

- **Швидкодія обробки даних:** час порівняння одного пакету транзакцій із багатотисячною санкційною базою даних за допомогою нечіткого пошуку не повинен викликати помітних затримок у роботі інтерфейсу, забезпечуючи комфортну роботу оператора в режимі реального часу.

Сформовані функціональні вимоги дозволяють перейти до наступного етапу роботи — проектування архітектури системи, розробки логічної структури бази даних та реалізації відповідних програмних модулів.

Висновки до першого розділу

У першому розділі кваліфікаційної роботи проведено аналіз предметної області та визначено ключові інформаційні потреби підрозділів внутрішнього контролю щодо фінансового моніторингу. Огляд існуючих платформ засвідчив, що їх використання для потокової перевірки транзакцій є неоптимальним через ризики порушення інформаційної безпеки. Це обґрунтовує необхідність створення власної локальної системи, до якої сформовано комплекс функціональних вимог. Встановлено, що для забезпечення конфіденційності та точності верифікація транзакцій має базуватися на моделях нечіткої логіки, а сам застосунок повинен мати зручний та зрозумілий інтерфейс.

РОЗДІЛ 2. ІНФОЛОГІЧНЕ МОДЕЛЮВАННЯ ТА РОЗРОБКА АЛГОРИТМІВ ФУНКЦІОНУВАННЯ СИСТЕМИ

2.1 Аналіз архітектури та функціональне моделювання системи

Проектування інформаційної системи вимагає комплексного підходу до моделювання її архітектури, що дозволяє відобразити взаємозв'язки між процесами обробки даних та користувачами. Для опису загального функціонування системи було побудовано функціональну модель у стандарті IDEF0.

IDEF0 – методологія, яка являє собою стандарт моделювання, призначений для графічного опису бізнес-процесів. Головна мета цієї методології полягає у побудові схеми, яка наочно демонструє функціональний склад системи та логічні зв'язки між операціями через рух потоків даних і ресурсів. Має позиції вхідних даних, вихідних результатів, керуючих впливів та механізмів виконання [4].

Функціональну модель системи (IDEF0) показано на рисунку 2.1

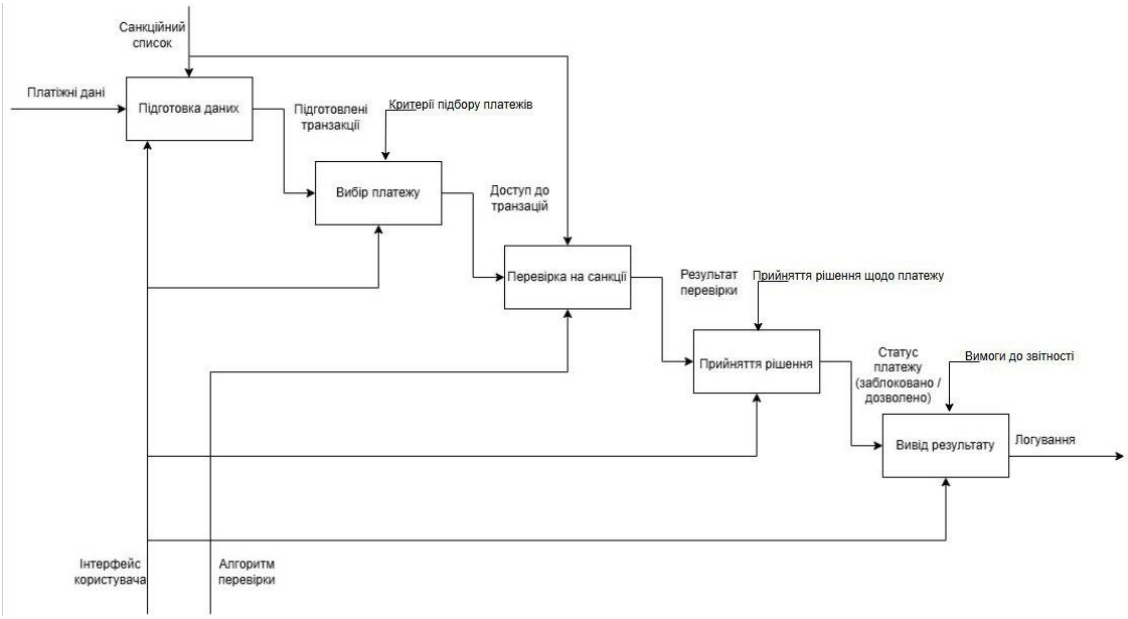

Рис. 2.1 - Функціональна модель системи (IDEF0)

Ця модель відображає основні інформаційні потоки: вхідними даними є запити на перевірку транзакцій та актуальні санкційні списки, а вихідними — результати перевірки з відповідними статусами ризику. Процес обробки охоплює ключові етапи: підготовку даних, вибір платежу, алгоритмічну перевірку та прийняття остаточного рішення.

Для деталізації меж системи та визначення ролей користувачів було розроблено діаграму прецедентів.

Діаграма прецедентів – візуально зображає різноманітні сценарії взаємодії між акторами і прецедентами та описує функціональні аспекти системи [5].

На рисунку 2.2 зображено діаграму прецедентів інформаційної системи перевірки платежів.

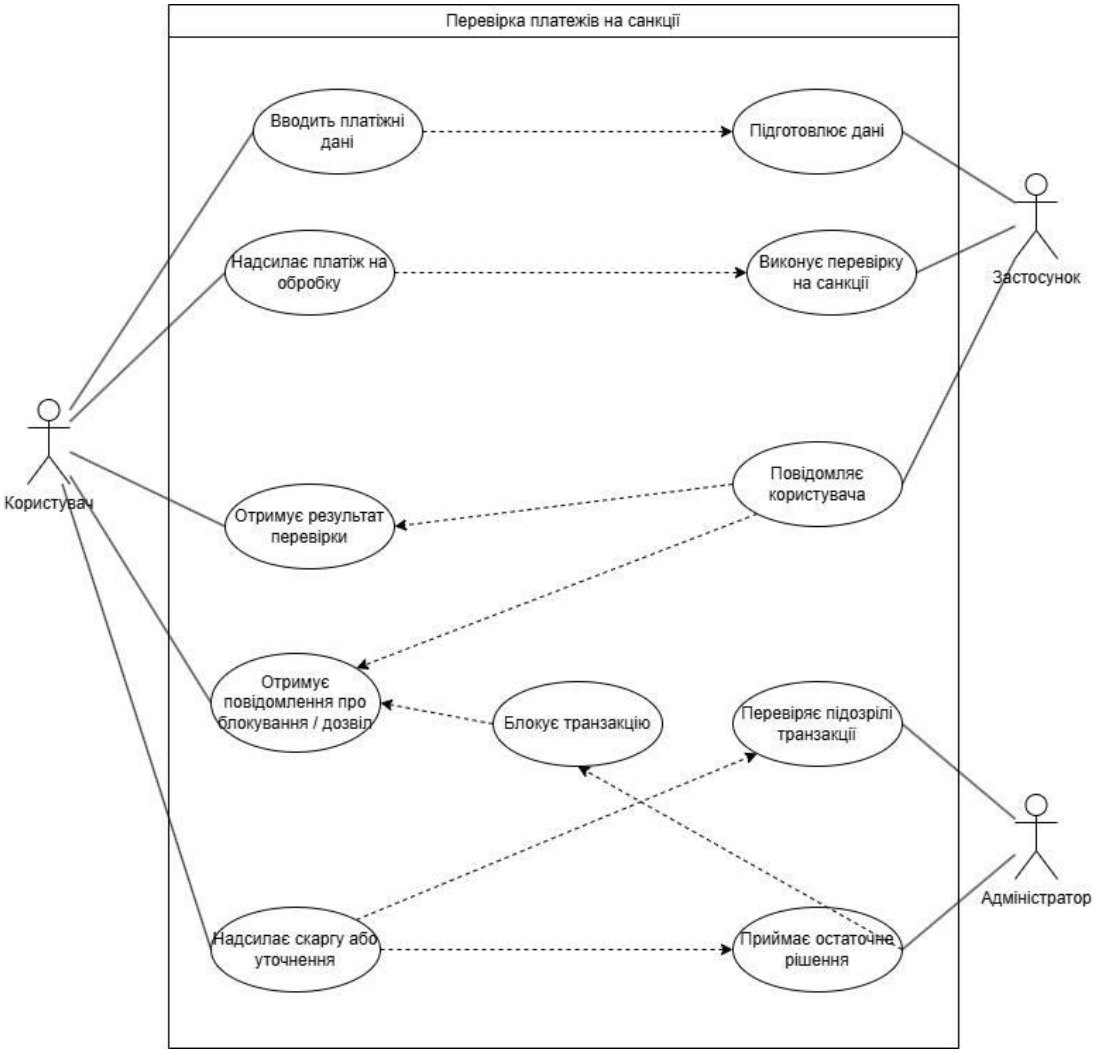

Рис. 2.2 - Діаграма прецедентів інформаційної системи перевірки платежів

Розроблена модель відображає розподіл функціональних обов'язків та взаємодію між трьома ключовими акторами системи:

1. Користувач (Оператор): виступає головним ініціатором перевірки. Він здійснює введення платіжних даних у систему та надсилає платіж на обробку. Після завершення автоматичного аналізу користувач отримує фінальний результат перевірки та відповідне сповіщення про блокування або дозвіл на проведення

операції. У разі виникнення спірних питань оператор має можливість надіслати скаргу чи уточнення для додаткового перегляду.

2. Застосунок: виконує роль автоматизованого ядра системи. Він бере на себе рутинні операції: попередньо підготовлює та очищує вхідні дані, безпосередньо виконує алгоритмічну перевірку інформації за санкційними реєстрами РНБО та миттєво сповіщає користувача про знайдені збіги.

3. Адміністратор: підключається до процесу перевірки як спеціаліст вищого рівня у випадках, коли Оператор не може прийняти остаточне рішення самостійно (наприклад, при виникненні технічних збоїв, неоднозначності вхідних даних або спірних результатах перевірки). Адміністратор детально аналізує такі підозрілі операції, здійснює ручну перевірку через прямий доступ до бази санкційних списків і виносить фінальний вердикт щодо блокування або дозволу на проведення платежу.

Головним функціональним елементом у цій структурі є процес «Виконує перевірку на санкції», логіка якого базується на використанні спеціальних математичних методів обробки текстової інформації, що будуть детально розглянуті у наступному підрозділі.

2.2 Алгоритм нечіткого пошуку та його математична модель

Функціонування системи моніторингу базується на алгоритмічному аналізі текстової інформації, що дозволяє виявляти санкційні ризики навіть за наявності невідповідності у вхідних даних (друкарські помилки, варіанти транслітерації, різна послідовність ініціалів). Для вирішення цього завдання застосовано математичну модель нечіткого пошуку (нечітка логіка), яка забезпечує гнучкість та надійність процесу прийняття рішень.

Нечітка логіка - це розділ математичної логіки, який оперує поняттям ступеня істинності та дозволяє логічним змінним приймати будь-які дійсні значення у безперервному діапазоні [0, 1]. Використовується для моделювання процесів прийняття рішень в умовах невизначеності, неповної інформації або нестрогої відповідності критеріям [6].

1. Математична основа (Метрика схожості). Алгоритмічну основу реалізації складає використання метрики подібності рядків, що реалізована через клас *SequenceMatcher* (стандартна бібліотека *difflib*). Метод базується на обчисленні співвідношення довжини спільних підпослідовностей до загальної довжини порівнюваних рядків.

Коефіцієнт схожості $R \in [0,1]$, розраховується за формулою:

$$R = \frac{2 * M}{N_1 + N_2}$$

де:

- M - кількість збіжних символів;

- N_1 та N_2 - довжини порівнюваних рядків (назва з транзакції та назва з бази санкційних списків відповідно).

2. Модель нечіткої логіки. Для формалізації процесу прийняття рішення в системі впроваджено модель нечіткої логіки, що базується на визначенні лінгвістичної змінної S («Ступінь подібності»), яка приймає значення з універсуму $U = [0, 100]$. Множина термів для даної змінної визначається як:

$$T(S) = \{\text{"Безпечна"}, \text{"Підозріла"}\}$$

Функції належності для визначення терму базуються на пороговому значенні $a = 85\%$ (0.85 у програмній реалізації):

- Для терміну «Безпечна»: $\mu_{\text{безпечна}}(x) = 1$, якщо $x < 85$, та 0 в іншому випадку.

- Для терміну «Підозріла»: $\mu_{\text{підозріла}}(x) = 1$, якщо $x \geq 85$, та 0 в іншому випадку.

Графічне представлення правил визначення лінгвістичної змінної та розподіл функцій належності наведено на рисунку 2.3.

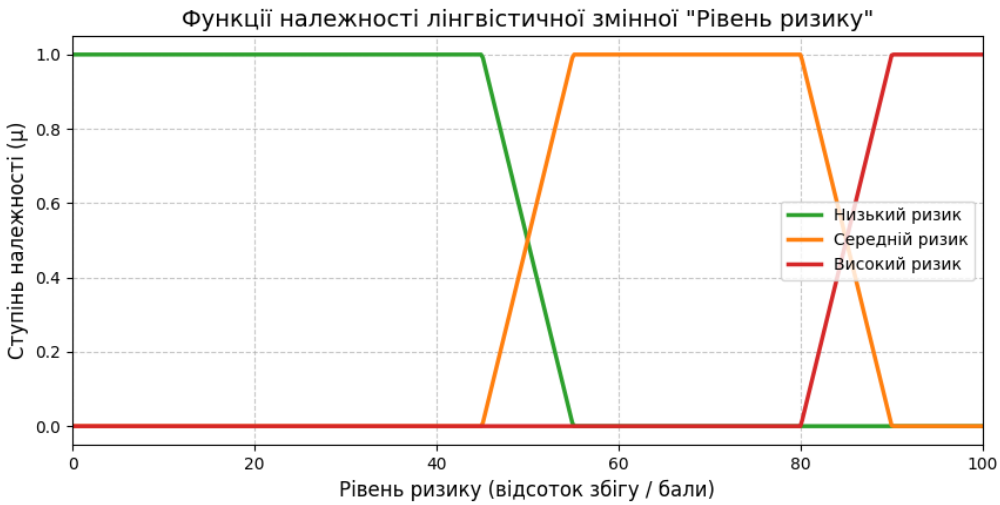

Рис. 2.3 - Графічне представлення функцій належності лінгвістичної змінної

Як відображено на графіку функцій належності, універсум значень поділяється на три інтервали: низький рівень ризику охоплює діапазон від 0 до 50, середній — від 50 до 85, а високий — від 85 до 100.

3. Правила прийняття рішень. Логіка функціонування алгоритму базується на продукційних правилах вида «ЯКЩО — ТО»:

- ЯКЩО (ступінь подібності $x \geq 85\%$), ТО (статус транзакції = «Підозріла») та здійснюєть ретельна перевірка транзакції

- ЯКЩО (ступінь подібності $x < 85\%$), ТО (статус транзакції = «Безпечна») та дозвіл на проведення операції.

Вибір порогу 85% є компромісним рішенням, що дозволяє мінімізувати ймовірність пропуску ризикової операції при збереженні високої швидкості роботи системи. Значення нижче 85% призводить до надмірної кількості хибних спрацьовувань, що перевантажує адміністратора великою кількістю несуттєвих збігів. Значення вище 85% створює ризик пропускання навмисних викривлень у даних (наприклад, використання візуально схожих символів кирилиці та латиниці).

Програмну реалізацію описаної математичної моделі нечіткого пошуку виконано мовою Python. Фрагмент коду, що реалізує логіку порівняння та перевірку порогового значення $a = 85\%$, наведено у додатках (див. Додаток В)

Загальний алгоритм роботи системи, що включає етап ініціалізації застосунку, вибір транзакції, безпосередній запуск математичної перевірки та прийняття фінального рішення на основі виявлених збігів, детально відображено на діаграмі станів, яку винесено у додатки (див. додаток А). Модель демонструє повний цикл обробки даних: у разі перевищення порогу чутливості ініціюється блокування транзакції, а за відсутності ризиків система повертається до стану очікування нової партії платіжних документів.

2.3 Моделювання бази даних та програмна реалізація системи

Для забезпечення надійного зберігання, швидкої обробки та цілісності інформації в системі моніторингу було спроектовано реляційну базу даних. Враховуючи вимоги до автономності та швидкодії настільного застосунку, як систему керування базами даних обрано SQLite.

SQLite - це компактна вбудована реляційна база даних з відкритим кодом, використовується в додатках та системах, де потрібно організувати зберігання даних [7].

Розроблена інфологічна модель (ER-діаграма), що ілюструє ключові сутності предметної області та зв’язки між ними, представлена на рисунку 2.4.

Рис. 2.4 - Інфологічна модель (ER-діаграма) бази даних системи моніторингу

Архітектура бази даних складається з шести взаємопов'язаних таблиць, кожна з яких виконує певну функціональну роль у процесі перевірки та аналізу транзакцій:

1. Таблиця users (Користувачі): Призначена для зберігання облікових записів персоналу. Містить ідентифікатори, логіни (*username*), паролі, повні імена та рівні доступу (*role* — Оператор або Адміністратор). Це забезпечує розмежування прав у системі.

2. Таблиця sanctions_list (Санкційні списки): Центральний довідник системи, що містить актуальні дані про підсанкційних осіб. Включає оригінальні імена (*name*), їх транслітерацію (*translit_name*), дати народження, громадянство, податкові номери (*tax_id*) та деталі щодо термінів і умов діючих санкцій.

3. **Таблиця transactions_list (Реєстр транзакцій):** Зберігає всю інформацію про фінансові операції, що надходять на перевірку. Поля включають дані відправника (*sender_name*), отримувача (*receiver_name*), суму, валюту,

призначення платежу (*details*) та поточний статус обробки (*status* — Pending, Approved, Blocked).

4. **Таблиця audit_logs (Журнал аудиту)**: Критично важливий елемент для систем фінансового моніторингу. Ця таблиця фіксує кожну дію користувачів щодо ухвалення рішень за транзакціями. Вона пов'язана зовнішніми ключами (*Foreign Keys*) з таблицями transactions_list (*trans_id*) та users (*user_id*), що дозволяє точно відстежити, який оператор і коли заблокував або пропустив конкретний платіж.

5. **Таблиці global_stats та system_config (Службові дані):** Використовуються для збереження глобальних метрик системи (загальна кількість оброблених, заблокованих та дозволених транзакцій) та конфігураційних налаштувань, таких як дата останнього оновлення баз та поріг чутливості алгоритму нечіткого пошуку (*fuzzy_threshold*).

Для деталізації покрокового виконання процесу обробки платіжних документів та візуалізації обміну повідомленнями між програмними об'єктами було побудовано діаграму послідовності, яку винесено у додатки (див. додаток Б). На цій моделі відображаються ключові компоненти системи, необхідні для виконання передбаченого сценарію верифікації, та повідомлення, які вони передають один одному у хронологічній послідовності [8].

Відповідно до розробленої моделі, сценарій взаємодії розділено на три логічні блоки:

1. Завантаження списку: При запуску програми інтерфейс автоматично отримує з бази даних перелік транзакцій та виводить їх на екран оператора.

2. Вибір платежу: Оператор обирає конкретну транзакцію зі списку для проведення аналізу.

3. Перевірка на санкції: Програма завантажує базу підсанкційних осіб, виконує алгоритм нечіткого порівняння імен та показує результат на екрані.

4. Прийняття рішення: Якщо система знайшла збіг, оператор підтверджує ризик і натискає кнопку «Блокувати».

5. Оновлення бази: Програма змінює статус цієї транзакції у базі даних на «Заблоковано» та виводить користувачеві повідомлення про успішне завершення дії.

Побудована інфологічна та поведінкова архітектура повністю описує внутрішню будову та логіку роботи застосунку. Це забезпечує надійне підґрунтя для програмної реалізації всіх заявлених функцій системи та гарантує її відповідність вимогам технічного завдання.

Висновки до другого розділу

У другому розділі було проведено комплексне проектування інформаційної системи перевірки платежів. За допомогою методології IDEF0 та діаграми прецедентів деталізовано логіку роботи застосунку, визначено інформаційні потоки та чітко розмежовано ролі ключових акторів (Оператор, Застосунок, Адміністратор).

Важливим етапом стала розробка математичної моделі на основі нечіткої логіки та метрики подібності рядків. Цей алгоритм дозволяє ефективно виявляти ризикові фінансові операції навіть за наявності друкарських помилок у вхідних даних. Теоретично обґрунтовано оптимальний поріг чутливості на рівні 85%, що забезпечує ідеальний баланс між надійністю блокування санкційних платежів та мінімізацією хибних спрацьовувань.

Крім того, спроектовано надійну інфологічну архітектуру бази даних (SQLite) для зберігання інформації та ведення журналу аудиту. Побудовані поведінкові UML-моделі наочно побачити динаміку взаємодії компонентів під час обробки документів.

РОЗДІЛ 3. ОПИС ІНТЕРФЕЙСУ ТА КЕРІВНИЦТВО КОРИСТУВАЧА РОЗРОБЛЕНОЇ СИСТЕМИ

3.1 Опис графічного інтерфейсу користувача та засобів програмної реалізації

Програмну реалізацію розробленої інформаційної системи моніторингу виконано об’єктно-орієнтованою мовою програмування Python. Для створення графічного інтерфейсу користувача було використано сучасну бібліотеку PyQt6.

PyQt — це бібліотека Python для створення програм із графічним інтерфейсом користувача за допомогою набору інструментів Qt.

Вибір PyQt6 зумовлений його багатим набором функцій, високою продуктивністю, універсальності застосування та архітектурною здатністю ефективно відокремлювати бізнес-логіку системи від візуального інтерфейсу.

Графічний інтерфейс системи спроектовано з урахуванням принципів ергономіки та зручності використання . Головна мета інтерфейсу - мінімізувати навантаження на оператора та забезпечити швидкий доступ до ключових функцій: перевірки транзакцій, перегляду списків та аналізу результатів.

Першим етапом взаємодії користувача з інформаційною системою є вікно авторизації (рис. 3.1), яке забезпечує захист даних від несанкціонованого доступу.

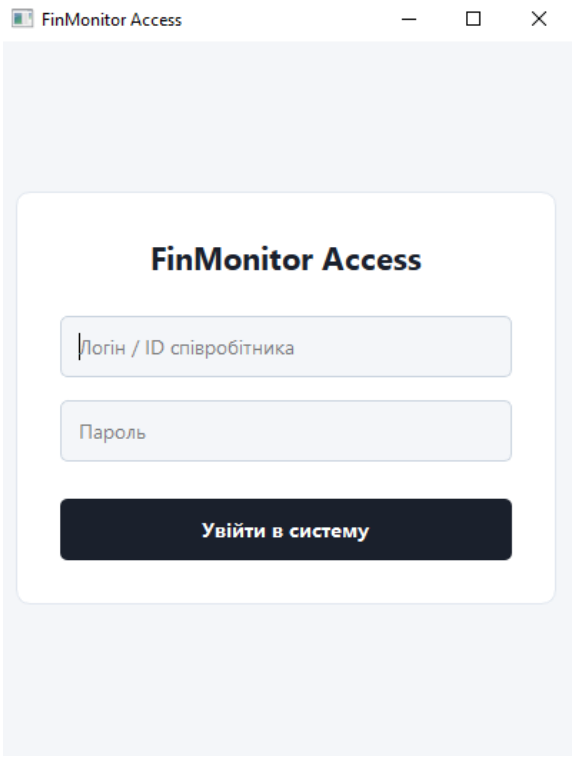

Рис. 3.1 – Вікно авторизації користувача

Інтерфейс містить поля для введення логіна і пароля та кнопку «Увійти». Для безпеки символи пароля маскуються під час введення. У разі успішної ідентифікації програма автоматично відкриває головне вікно, а при введенні некоректних даних виводить повідомлення про помилку.

Після успішної авторизації користувач переходить до головного робочого вікна системи (рис. 3.2), яке є основним вікном для управління процесами фінансового моніторингу.

Рис. 3.2 – Головний екран системи FinMonitor

Інтерфейс вікна побудований за принципом панелі навігації зліва та робочої області справа. Ліва бічна панель забезпечує швидкий доступ до основних розділів системи: «Головна панель», «Всі транзакції», «Санкційні списки» та «Журнал аудиту», а також містить кнопку «Налаштування» та відображення даних поточного користувача (Іванчук Нікіта, Оператор).

Основну частину робочої області займає розділ «Огляд ситуації», де за допомогою інформаційних карток представлено ключові метрики системи в режимі реального часу:

- Очікують (Зараз): кількість транзакцій, що потребують перевірки;
- Заблоковано (Всього): загальна кількість виявлених та заблокованих ризикових операцій;
- Успішно (Всього): кількість транзакцій, що успішно пройшли верифікацію;

- Оброблено (Всього): загальна кількість оброблених системою запитів.

Наступним ключовим компонентом системи є модуль «Всі транзакції» (рис. 3.3), призначений для детального аналізу та управління фінансовими операціями.

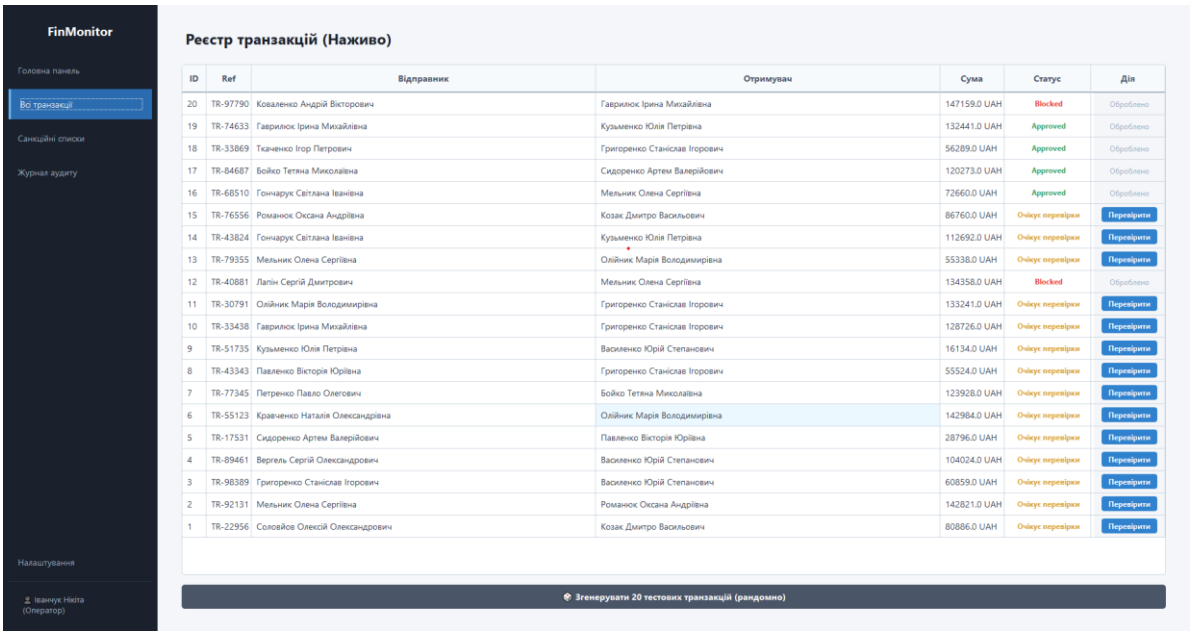

Рис. 3.3 - Інтерфейс модуля «Всі транзакції»

Робоча область модуля представлена у вигляді інтерактивної таблиці, що містить повний перелік платежів з інформацією про відправника, отримувача, суму, валюту та поточний статус обробки. Кожен рядок таблиці відображає актуальний стан транзакції, дозволяючи оператору візуально ідентифікувати платежі, що очікують на перевірку, або ті, що вже були заблоковані.

Для безпосереднього аналізу ризиковості платіжних операцій у системі передбачено інтерфейс перевірки транзакції (рис. 3.4).

Це вікно відкривається при виборі конкретного платежу і містить деталізовану інформацію про операцію: дані відправника, отримувача, суму та призначення платежу. Центральним елементом інтерфейсу є блок з результатами роботи алгоритму нечіткого пошуку, де відображається відсоток схожості між даними транзакції та записами у санкційних реєстрах.

Для завершення операції користувач використовує кнопки керування «Заблокувати» або «Дозволити», після натискання яких статус транзакції оновлюється в базі даних, а дія фіксується у журналі аудиту. Така організація інтерфейсу дозволяє оператору швидко приймати обґрунтовані рішення,

спираючись на точні аналітичні дані, отримані в результаті автоматизованого аналізу.

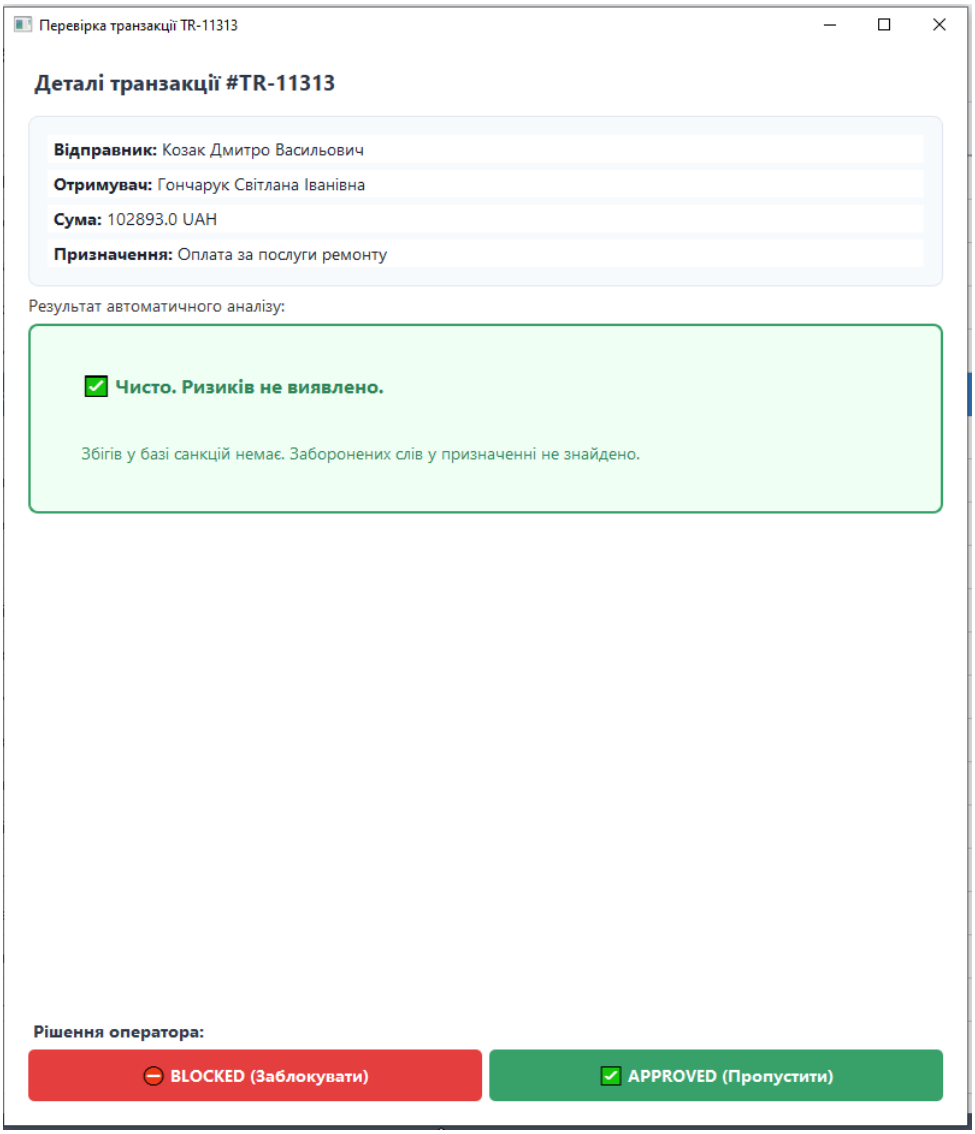

Рис. 3.4 - Інтерфейс перевірки транзакції

Для роботи з нормативно-правовою базою в системі реалізовано модуль «Санкційні списки» (рис. 3.5), який виступає основним довідником для проведення верифікації.

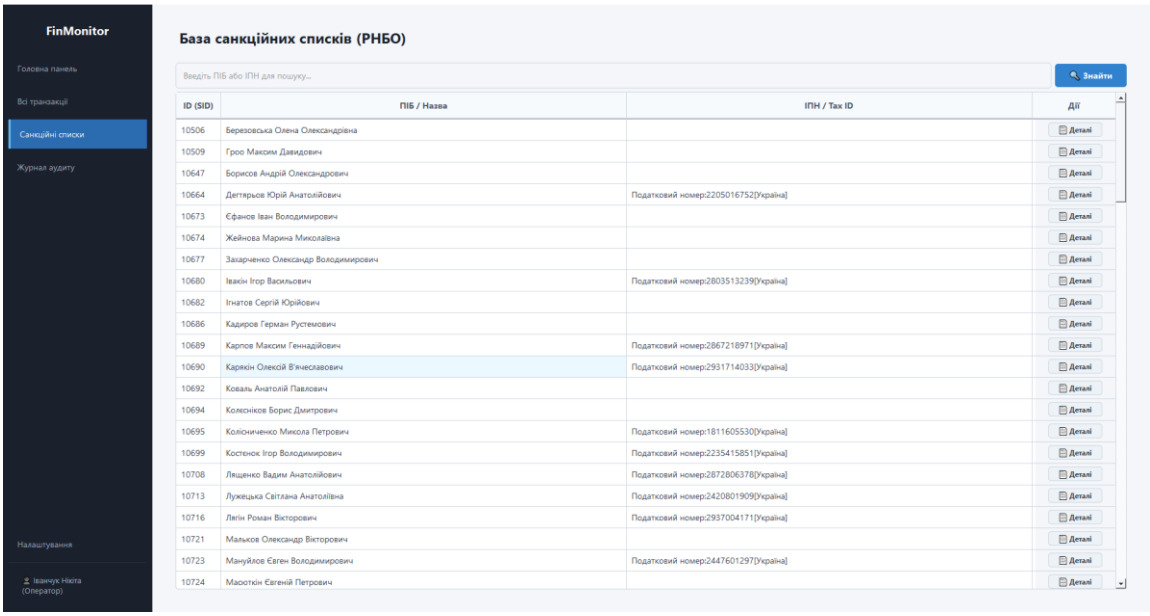

Рис. 3.5 – Інтерфейс модуля «Санкційні списки»

Робоча область модуля містить структуровану таблицю з даними про підсанкційних осіб. Інтерфейс має функцію швидкого пошуку, що дозволяє оператору миттєво знайти особу в базі за ключовими даними, що є критично важливим для оперативного контролю. Кожен запис у таблиці супроводжується кнопкою «Деталі», яка надає доступ до розширеної інформації про підстави та терміни застосування санкцій. На рисунку 3.6 наведено модуль детальної інформації підсанкційної особи.

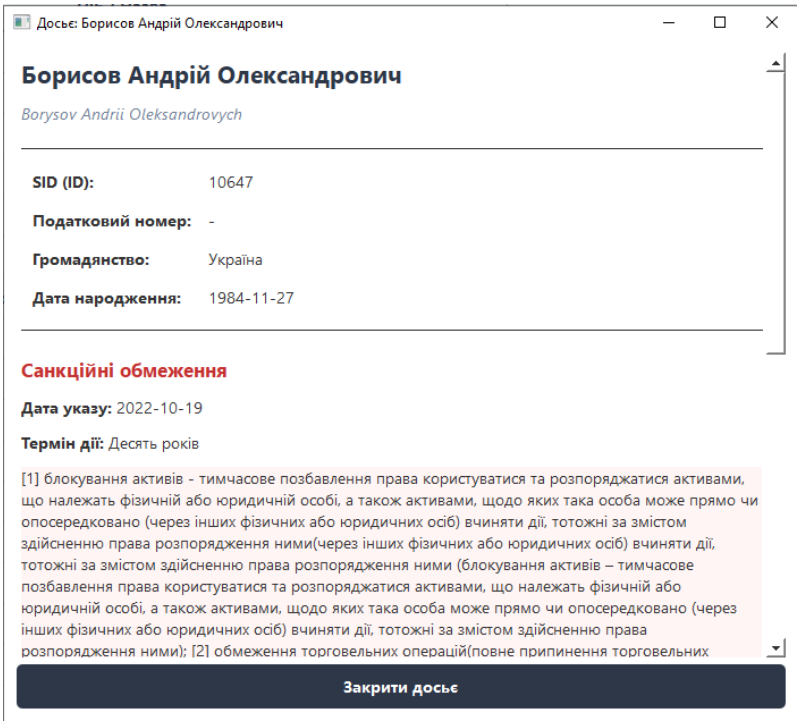

Рис. 3.6 - Інтерфейс модуля детальної інформації підсанкційної особи

Даний інтерфейс реалізовано у форматі інформаційної картки, яка систематизує всі доступні дані про суб’єкт санкцій. У вікні відображаються такі поля, як повне ім'я, дата народження, громадянство, ідентифікаційний номер та категорія обмежень.

Для забезпечення прозорості та контролю над усіма діями користувачів у системі передбачено модуль «Журнал аудиту» (рис. 3.7).

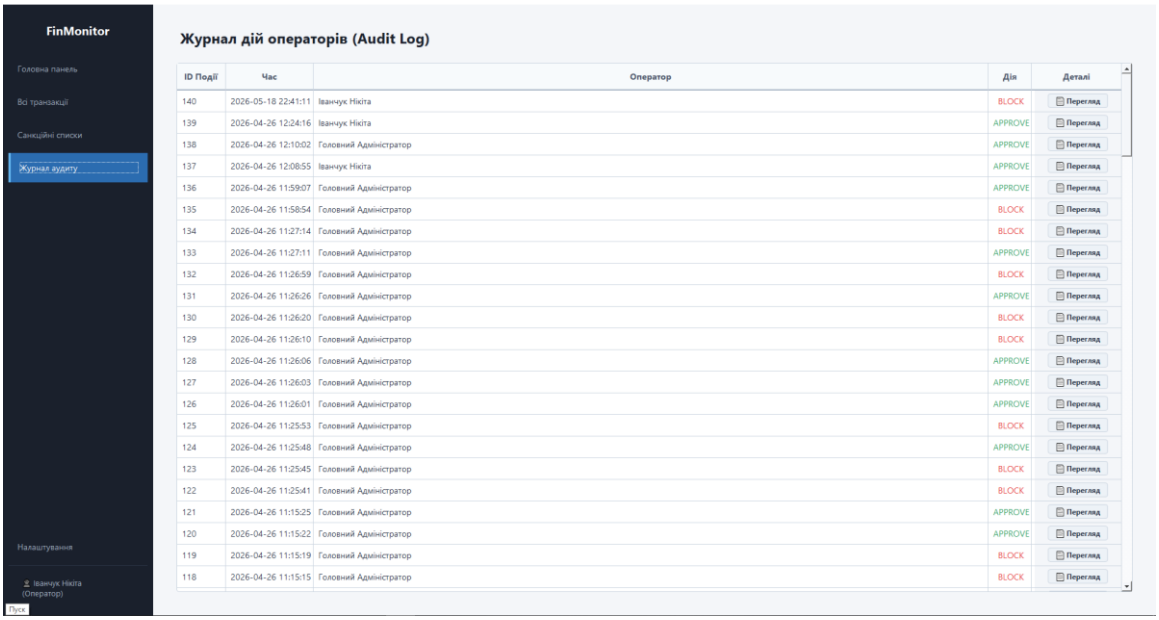

Рис. 3.7 – Інтерфейс модуля «Журнал аудиту»

Цей модуль реалізує функцію безперервного моніторингу, автоматично реєструючи кожну зміну статусу транзакції. Таблиця журналу містить унікальний ідентифікатор події, точний час виконання операції, логін оператора, який прийняв рішення, та тип виконаної дії. Наявність кнопки «Перегляд» у кожному рядку дозволяє адміністратору миттєво отримати детальну інформацію про конкретний інцидент. Такий рівень деталізації дій забезпечує високий рівень підзвітності в системі.

3.2 Порядок розгортання та налаштування програмного забезпечення

Оскільки розроблена система є спеціалізованим корпоративним інструментом для здійснення фінансового моніторингу, її розгортання на робочих станціях операторів виконується централізовано технічним відділом підприємства. Програмне забезпечення постачається у вже налаштованому вигляді, що повністю

виключає необхідність виконання будь-яких інсталяційних дій чи конфігурації середовища з боку кінцевого користувача. Оператору достатньо здійснити запуск застосунку через відповідний ярлик на робочому столі.

Окрему увагу в системі приділено безпеці доступу: механізм самостійної реєстрації нових облікових записів користувачами відсутній. Створення облікового запису, надання прав доступу та управління ними здійснюється виключно адміністратором системи. При першому зверненні до робочого місця оператор отримує від адміністратора персональні дані для авторизації - логін та пароль, які забезпечують ідентифікацію особи в системі та дозволяють розпочати роботу з фінансовими документами.

3.3 Керівництво користувача системи

Нижче наведено методичні вказівки щодо експлуатації програмного комплексу. Документ орієнтований на фахівців служби фінансового моніторингу та охоплює повний цикл обробки транзакцій.

1. Авторизація в системі:

Робота розпочинається із запуску системи. У вікні входу (див. рис. 3.1) необхідно ввести особистий логін співробітника та пароль.

2. Моніторинг поточного стану:

Після успішної авторизації відкривається головна панель (див. рис. 3.2), де оператор повинен проаналізувати статистичні показники, якщо картка «Очікують перевірки» більший за нуль, це вимагає негайних дій.

3. Робота з транзакціями:

Для переходу до детальної обробки платежів необхідно перейти у розділ «Всі транзакції» через бокове меню (див. рис. 3.3), далі потрібно обрати транзакцію для аналізу та натиснути кнопку «Перевірити».

4. Перевірка та прийняття рішення:

У вікні детального перегляду (див. рис. 3.4) виконуються наступні дії:

1. Звірка реквізитів: перевірка відповідності даних відправника та призначення платежу внутрішнім правилам банку.

2. Аналіз ризиків:

- Якщо панель відображає повідомлення «Збіг знайдено» (червоний індикатор), необхідно ознайомитися з назвою санкційного списку та підставою.

- Якщо автоматичний пошук не виявив загроз, але транзакція здається підозрілою, слід виконати ручний пошук у розділі «Санкційні списки».

3. Фіксація рішення:

- Для блокування: натиснути червону кнопку «Заблокувати».

- Для пропуску: натиснути кнопку «Пропустити» виключно за умови впевненості, що збіг є помилковим (наприклад, збіг імен за відсутності інших ідентифікаторів).

5. Завершення роботи

Після закінчення зміни або при залишенні робочого місця оператор зобов'язаний вийти з системи шляхом натискання кнопки «Вихід» для запобігання несанкціонованому доступу до даних.

Адміністратор системи відповідає за підтримку актуальності довідкової інформації. Процедура оновлення санкційних списків виконується шляхом заміни файлу даних та подальшої синхронізації з базою даних системи. Для виконання оновлення необхідно виконати наступні дії:

1. Завантаження даних: Відвідати офіційний портал Ради національної безпеки і оборони України (РНБО) та завантажити актуальний перелік підсанкційних осіб у форматі CSV.

2. Підготовка файлу: Перемістити завантажений файл у кореневу директорію програмного комплексу та змінити його назву на sanctions.csv, замінивши попередню версію файлу.

3. Синхронізація з БД: Запустити застосунок та авторизуватися з правами адміністратора.

- Перейти до розділу «Налаштування»

- Натиснути кнопку «СПОВІСТИТИ» (див. рис. 3.8), яка ініціює процес зчитування нового файлу та автоматичне оновлення внутрішньої бази даних системи.

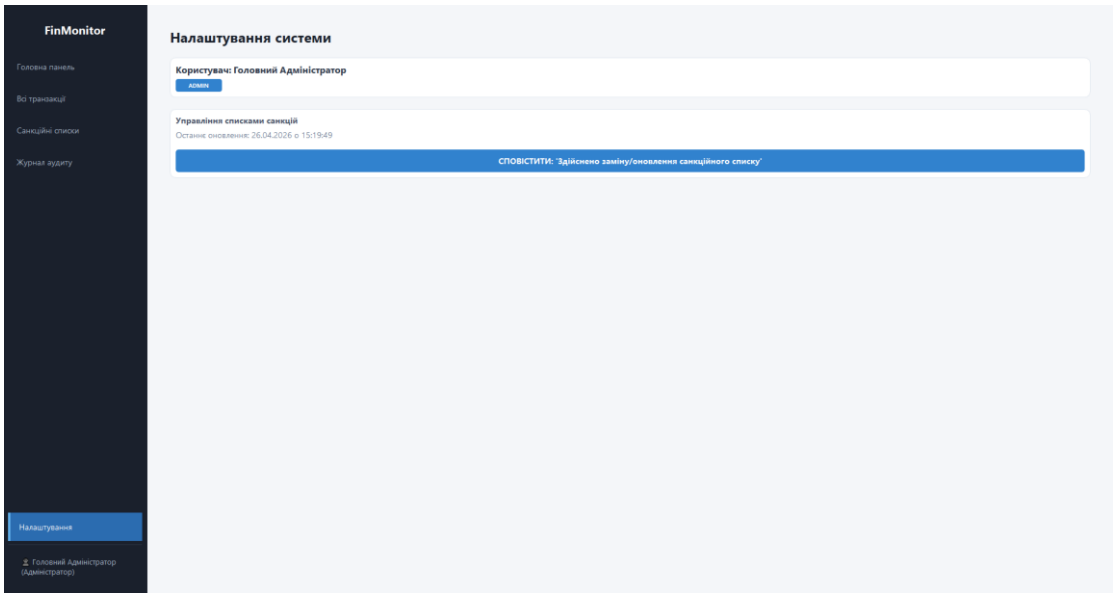

Рис. 3.8 - Елементи керування у розділі «Налаштування»

Після завершення процесу система автоматично повідомить про успішне оновлення бази, після чого нові дані стануть доступними для алгоритмів нечіткого пошуку та використання операторами.

3.4 Тестування системи

Для підтвердження коректності роботи розробленої інформаційної системи та алгоритму нечіткого порівняння було проведено контрольне тестування. З метою максимального наближення умов експерименту до реального робочого процесу служби фінансового моніторингу було попередньо згенеровано тестову вибірку, що складається з 20 транзакцій.

Цей масив даних цілеспрямовано включає як повністю безпечні платежі, так і ризикові. Для формування ризикових транзакцій використовувалися реальні відомості про осіб із бази санкційних списків РНБО, а також додавалися специфічні підозрілі призначення платежу. Такий підхід дозволив комплексно та об'єктивно перевірити чутливість алгоритмів системи до різних типів фінансових загроз.

Процес експериментального тестування було розділено на три послідовні етапи:

Етап 1. Перевірка безпечної транзакції

Оператор ініціював перевірку першої транзакції зі списку очікування. Після натискання кнопки перевірки, система передала реквізити платежу до модуля нечіткого пошуку. Результат роботи алгоритму показав відсутність схожості із санкційним списком. На рисунку 3.9 проведено тестову перевірку безпечної транзакції.

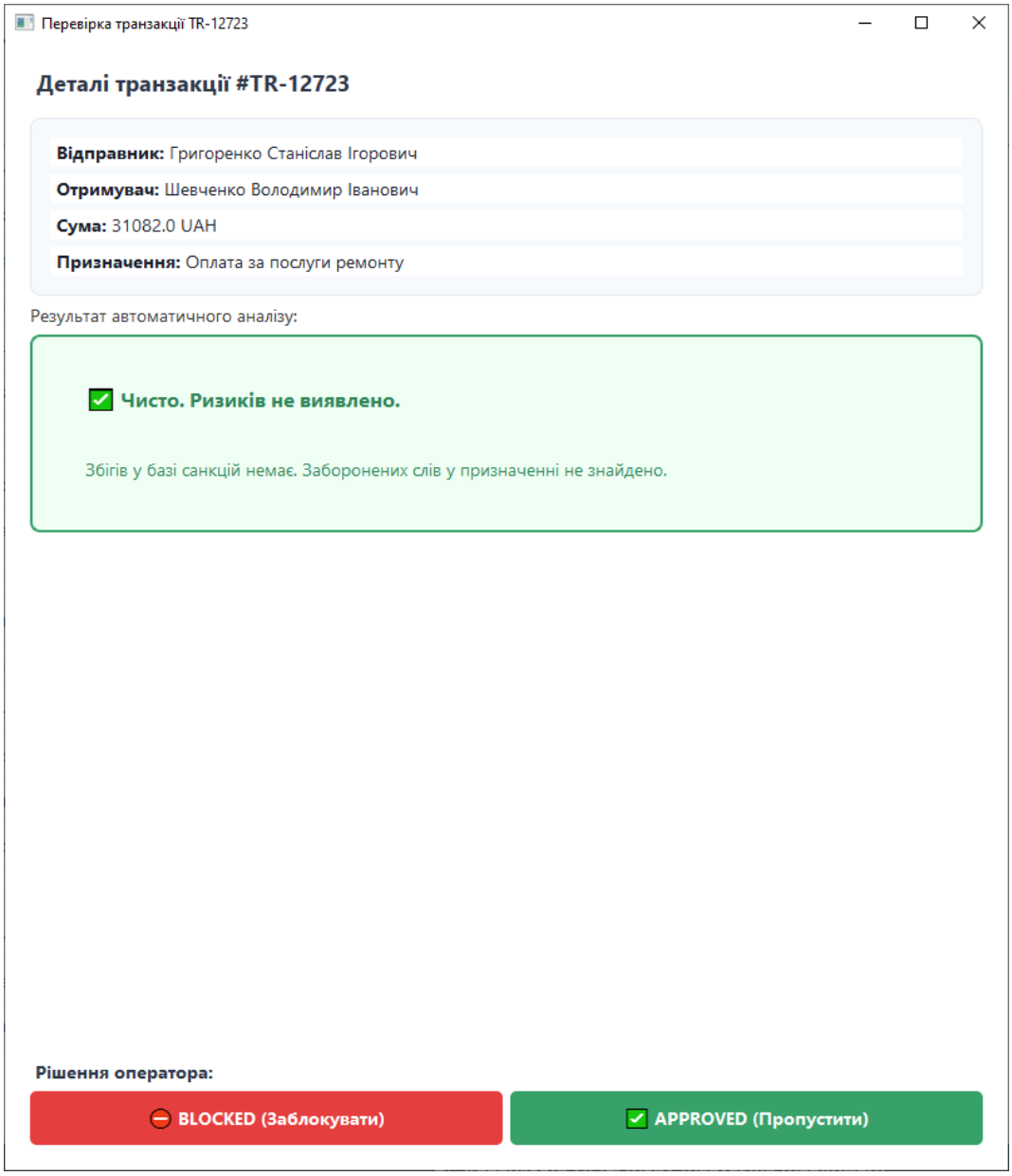

Рис. 3.9 – Тестова перевірка безпечної транзакцій

Інтерфейс системи не видав жодних попереджень про ризик. Оператор підтвердив безпечність операції, після чого статус транзакції у базі даних було успішно змінено на «Дозволено». На рисунку 3.10 показано зміну статусу транзакції.

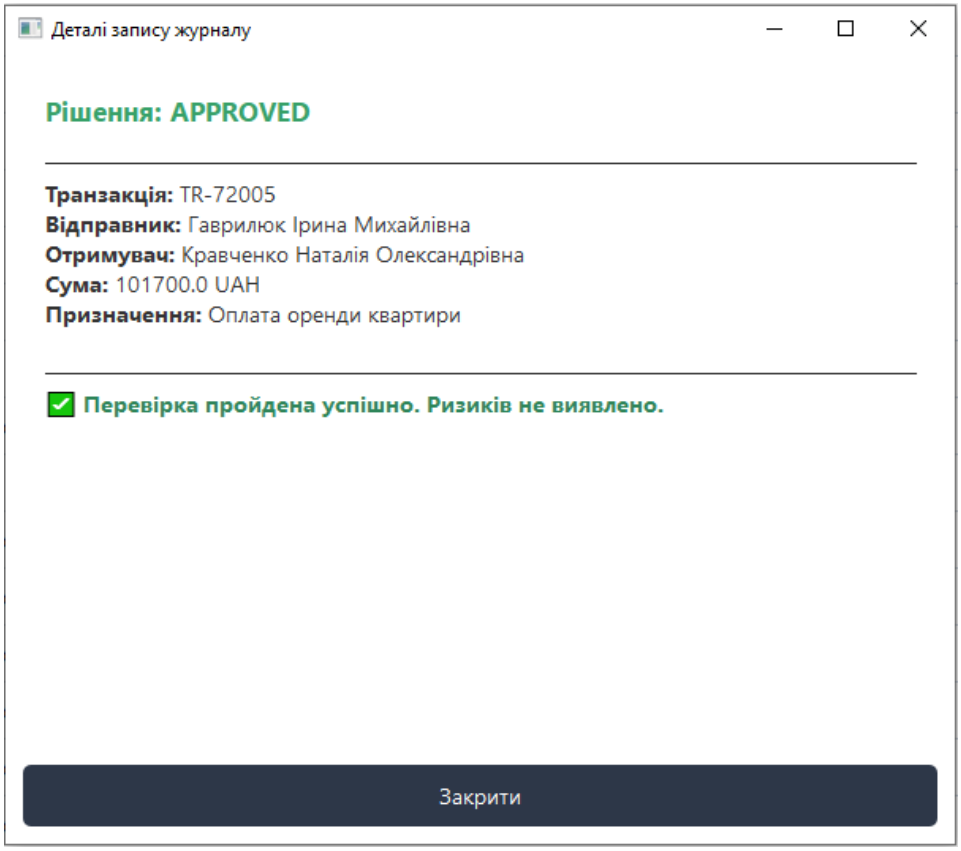

Рис. 3.10 – Зміна статусу безпечної транзакції на «Допущено»

Етап 2. Виявлення та блокування ризикової транзакції

Наступним кроком оператор обрав для аналізу транзакцію з категорії «брудних». Під час виконання перевірки система продемонструвала комплексну реакцію на загрозу. По-перше, алгоритм нечіткого пошуку виявив високий ступінь подібності імені учасника транзакції із записом у санкційному реєстрі. По-друге, система додатково зреагувала на наявність критичних стоп-слів у полі «Призначення платежу», ідентифікувавши його як потенційно небезпечне. На екрані детальної інформації миттєво з'явилося відповідне попередження (червоний індикатор) із зазначенням відсотка збігу та інформацією про підсанкційну особу. Проаналізувавши надані системою дані, оператор переконався у наявності ризику та натиснув кнопку «Заблокувати». На рисунку 3.11 показано тестову перевірку ризикової транзакції.

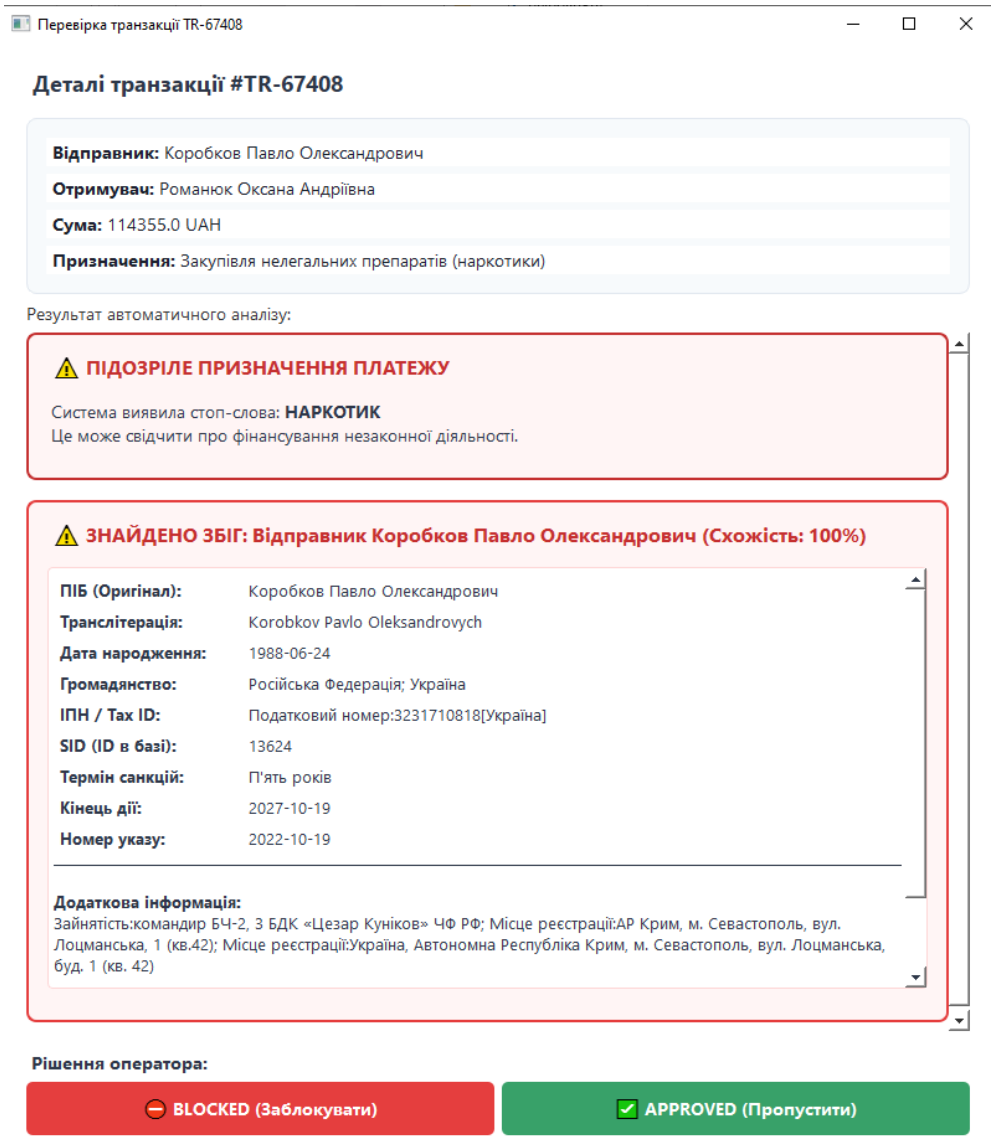

Рис. 3.11 – Тестова перевірка ризикової транзакції

Програма успішно оновила статус транзакції на «Заблоковано» і внесла відповідний запис до журналу аудиту із зазначенням часу та ідентифікатора оператора.

На рисунку 3.12 наведено зміну статусу ризикової транзакції на «Заблоковано», та інформація стосовно цієї транзакції.

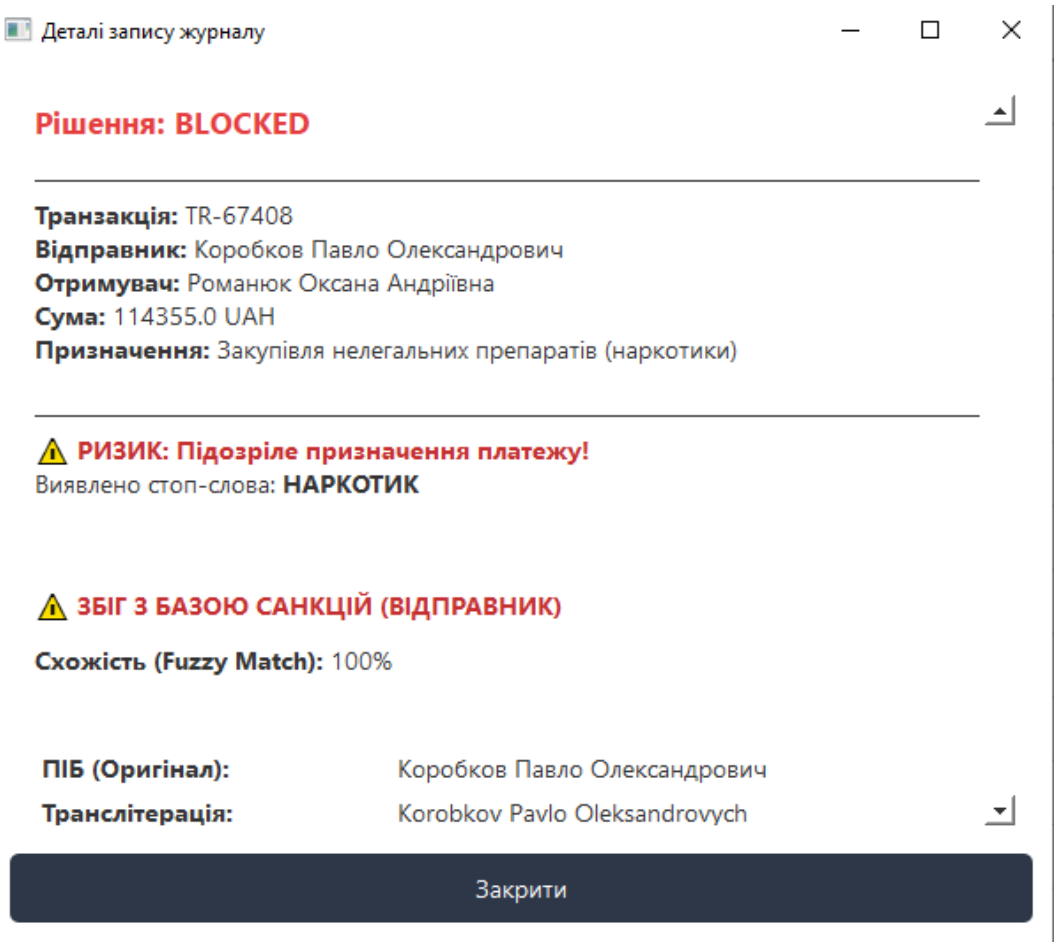

Рис. 3.12 – Зміна статусу ризикової транзакції на «Заблоковано»

Етап 3. Аналіз випадку хибного спрацьовування

Для перевірки гнучкості системи та оцінки ролі людського фактору в процесі внутрішнього пошуку на третьому етапі було проаналізовано транзакцію, реквізити якої містили частковий збіг із даними санкційного переліку.

Під час обробки платежу алгоритм нечіткого пошуку зафіксував схожість на рівні 88%, що дорівнює встановленому пороговому значенню чутливості. Такий високий відсоток збігу був зумовлений повною ідентичністю імені та по батькові учасника транзакції з анкетними даними особи із санкційного списку. Проте їхні прізвища були абсолютно різними. Система автоматично зреагувала на досягнення критичного порогу схожості й вивела на екран попередження про ризик.

На рисунку 3.13 наведено приклад хибної реакції системи перевірки.

Рис. 3.13 – Хибна реакція системи перевірки

Як видно на рисунку реакція системи перевірки сталась через схожість імені та по-батькові у відправника та підсанкційної особи, але прізвища у них різні.

Підсанкційна особа: КОЛОТ Андрій Вікторович

Відправник: Коваленко Андрій Вікторович

Детально дослідивши інформаційну картку та порівнявши прізвища відправника платежу та суб'єкта із санкційного реєстру, оператор класифікував це сповіщення як хибне спрацьовування. На основі виявленої невідповідності головного ідентифікатора (прізвища) оператор прийняв обґрунтоване рішення

пропустити платіж, натиснувши кнопку «Пропустити». Дана дія була автоматично зафіксована у журналі аудиту. Цей етап експерименту підтвердив важливість комбінованого підходу, за якого автоматизований аналіз алгоритму ефективно доповнюється фінальним рішенням спеціаліста.

Висновки до третього розділу

У третьому розділі було виконано практичну реалізацію клієнтської частини інформаційної системи та проведено перевірку її працездатності в умовах, наближених до реальної експлуатації.

По-перше, спроектовано та розроблено зручний графічний інтерфейс користувача за допомогою мови Python та бібліотеки PyQt6. Створений візуальний простір відповідає сучасним вимогам: він забезпечує швидкий доступ до ключових модулів (реєстру транзакцій, санкційних списків, журналу аудиту) та чітку візуалізацію ризиків, що значно знижує навантаження на оператора. Описано процедуру корпоративного розгортання системи із централізованим адмініструванням облікових записів для гарантування безпеки доступу.

По-друге, розроблено детальні експлуатаційні інструкції. Для операторів служби фінансового моніторингу складено покрокове керівництво з обробки платежів та прийняття рішень, а для адміністраторів — алгоритм оперативного оновлення нормативної бази (санкційних списків РНБО) через внутрішні налаштування системи.

По-третє, проведено комплексне експериментальне тестування алгоритмів на тестовій вибірці з 20 створених транзакцій. Перевірка успішно підтвердила коректність роботи системи у трьох ключових сценаріях:

- безперешкодний пропуск «чистих» безпечних платежів;
- миттєве виявлення та блокування ризикових операцій;
- ефективне опрацювання хибних спрацьовувань на рівні 88% схожості.

У підсумку, результати тестування доводять, що розроблена система є повністю працездатною, стабільною та готовою до використання. Вона успішно автоматизує рутинні процеси внутрішнього-контролю.

ВИСНОВКИ

У кваліфікаційній роботі успішно вирішено актуальне науково-практичне завдання щодо проектування та програмної реалізації автономної інформаційної системи моніторингу санкційних ризиків для підрозділів внутрішнього контролю. Аналіз предметної області довів, що використання зовнішніх онлайн-сервісів перевірки контрагентів несе суттєві ризики порушення банківської таємниці, що обґрунтувало необхідність створення повністю локального програмного продукту. Для забезпечення надійної роботи системи було проведено комплексне моделювання за допомогою методології IDEF0 та діаграм UML. На їх основі спроектовано оптимізовану базу даних під керуванням SQLite, ключовим елементом якої став захищений журнал аудиту, що гарантує фіксацію кожної дії оператора та забезпечує повну прозорість процесу прийняття рішень.

Науковою та алгоритмічною основою розробленого програмного комплексу стала математична модель, побудована на принципах нечіткої логіки та використанні метрики подібності рядків. Оскільки у фінансових документах часто трапляються друкарські помилки або різні варіанти транслітерації, реалізований алгоритм дозволяє ефективно виявляти часткові збіги анкетних даних із реєстрами РНБО. У ході дослідження підтверджено, що використання порогового значення чутливості на рівні 85% забезпечує ідеальний баланс, гарантуючи надійне виявлення загроз без перевантаження фахівців хибними спрацьовуваннями. Програмна реалізація клієнтської частини виконана мовою Python із залученням бібліотеки PyQt6, що дозволило створити зручний корпоративний інтерфейс із підтримкою швидкого оновлення нормативних баз через файли CSV.

Комплексне експериментальне тестування на вибірці з 20 контрольних транзакцій повністю підтвердило стабільність і коректність роботи розроблених алгоритмів. Система продемонструвала здатність безперешкодно пропускати безпечні операції, автоматично ідентифікувати ризикові платежі (завдяки комбінованому аналізу нечітких збігів та виявленню стоп-слів) і ефективно опрацьовувати складні випадки хибних спрацьовувань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. YouControl. *Рішення для бізнесу YouControl*. URL: https://youcontrol.com.ua (Дата звернення 11.06.2026)

2. Vkursi. *Спеціальна перевірка компаній та фізичних осіб*. URL: https://vkursi.pro (Дата звернення 11.06.2026)

3. Opendatabot. *Opendatabot*. URL: https://opendatabot.ua/en (Дата звернення 11.06.2026)

4. Визначення IDEF0. *Elib*. URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/Кондіус%202%20готовва/page9.html (Дата звернення 11.06.2026)

5. Визначення діаграма прецедентів. *blogspot*.URL: https://lvivqaclub.blogspot.com/2008/10/use-case-uml-diagram.html (Дата звернення 11.06.2026)

6. Визначення нечітка логіка. *StudFiles*. URL: https://studfile.net/preview/5176709/ (Дата звернення 11.06.2026)

7. Визначення SQlit. *Freehost*. URL: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-sqlite/ (Дата звернення 11.06.2026)

8. Визначення діаграма послідовності. *Maxzosim*. URL: https://www.maxzosim.com/sequence-diagrams/ (Дата звернення 11.06.2026)

9. Визначення PyQt6. *Mali.cx*. URL: https://wraneath.mali.cx.ua/articles/posibnik-pyqt.html (Дата звернення: 11.06.2026)

10. Іванчук Н. А. Технічні аспекти та програмна реалізація системи моніторингу санкційних ризиків // Концептуальні шляхи розвитку науки та освіти : матеріали XVIII Міжнар. наук.-практ. конф. (м. Львів, 29-30 травня 2026 р.). Львів, 2026. С. 96–97. URL: http://lviv-forum.inf.ua/save/2026/29-30.05/Збірник.pdf (Дата звернення: 11.06.2026)

11. Іванчук Н. А. Інформаційна система моніторінгу та аналізу санкційних ризиків у банківських операціях. *Інформаційні технології та моделювання систем* : зб. праць учасн. Всеукр. наук.-практ. конф. здобувачів вищої освіти і молодих вчених,

м. Житомир, 22 трав. 2026. Житомир : Поліський національний університет, 2026. С. 20-21 (Дата звернення: 11.06.2026)

12. Визначення діаграма станів. Guru99. URL: https://www.guru99.com/uk/statemachine-transition-diagram.html (Дата звернення: 11.06.2026)

13. Визначення діаграма класів. Mindonmap. https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram/ (Дата звернення: 11.06.2026)

14. Робота з модулем difflib у Python та метрики подібності. *Програмування українською*. URL: https://python.org.ua/difflib-module-sequence-matcher/ (дата звернення: 11.06.2026)

15. Закон України «Про запобігання та протидію легалізації (відмиванню) доходів, одержаних злочинним шляхом, фінансуванню тероризму...». *Верховна Рада України*. URL: https://zakon.rada.gov.ua/laws/show/361-20 (дата звернення: 11.06.2026)

16. Офіційні санкційні списки. *Рада національної безпеки і оборони України (РНБО)*. URL: https://www.rnbo.gov.ua/ua/sanctions/ (дата звернення: 11.06.2026)

17. Положення про здійснення банками фінансового моніторингу. *Національний банк України*. URL: https://bank.gov.ua/ua/legislation/Resolution_65 (дата звернення: 11.06.2026)

18. Діаграма діяльності (Activity diagram) в UML: призначення та побудова. *IT-освіта*. URL: https://it-osvita.com.ua/uml-activity-diagram/ (дата звернення: 11.06.2026)

19. Проектування баз даних: основи ER-діаграм. *Vertex Academy*. URL: https://vertex-academy.com/tutorials/uk/er-diagrams/ (дата звернення: 11.06.2026)

20. Патерн MVC (Model-View-Controller) у розробці графічних інтерфейсів. *Metanit Українською*. URL: https://metanit.com.ua/python/mvc-pattern/ (дата звернення: 11.06.2026)

21. Основи реляційних баз даних та SQL. *Prometheus*. URL: https://prometheus.org.ua/relational-databases-basics/ (дата звернення: 11.06.2026)

22. Читання та обробка CSV-файлів у Python за допомогою вбудованих бібліотек. *CodeUA*. URL: https://codeua.com/python/csv-parsing-tutorial/ (дата звернення: 11.06.2026)

23. Відстань Левенштейна та алгоритми нечіткого пошуку рядків. *Алгоритми та структури даних*. URL: https://algo.in.ua/levenshtein-distance-fuzzy-search/ (дата звернення: 11.06.2026)

24. Що таке KYC (Know Your Customer) у банківській сфері та чому це важливо. *Finance.ua*. URL: https://finance.ua/ua/saving/shcho-take-kyc (дата звернення: 11.06.2026)

25. Види та методи тестування програмного забезпечення. *QATestLab*. URL: https://qatestlab.ua/blog/software-testing-types/ (дата звернення: 11.06.2026)

26. Ризики використання зовнішніх API для збереження банківської таємниці. *CyberSecurity.ua*. URL: https://cybersecurity.ua/api-risks-banking-secrecy/ (дата звернення: 11.06.2026)

27. Переваги локальних Desktop-додатків над Web-рішеннями для корпоративного сектору. *IT-Arena*. URL: https://it-arena.ua/desktop-vs-web-enterprise/ (дата звернення: 11.06.2026)

28. Міжнародні стандарти FATF щодо боротьби з відмиванням грошей. *Державна служба фінансового моніторингу України*. URL: https://fiu.gov.ua/pages/mizhnarodne-spivrobitnictvo/fatf.html (дата звернення: 11.06.2026)

29. Оптимізація запитів та продуктивність у СУБД SQLite. *SQL.ua*. URL: https://sql.ua/sqlite-optimization-tips/ (дата звернення: 11.06.2026)

30.Моделювання архітектури: Діаграма розгортання (Deployment diagram). *TechEdu*. URL: https://techedu.com.ua/uml-deployment-diagram/ (дата звернення: 11.06.2026)

31. Нормалізація баз даних: перша, друга та третя нормальні форми. *Bionic University*. URL: https://bionic.com.ua/db-normalization-rules/ (дата звернення: 11.06.2026)

32. Впровадження систем комплаєнс-контролю в банках України. *Мінфін*. URL: https://minfin.com.ua/ua/banks/compliance-control/ (дата звернення: 11.06.2026)

33. Метрики подібності текстів у програмуванні та їх застосування. *Dev.ua*. URL: https://dev.ua/news/text-similarity-metrics (дата звернення: 11.06.2026)

34. Базові структури даних у Python: робота зі словниками та списками. *IT-Proger*. URL: https://itproger.com/ua/course/python/data-structures (дата звернення: 11.06.2026)

35. Принципи проектування ергономічних інтерфейсів користувача (UI/UX). *Projector*. URL: https://prjctr.com/mag/ui-ux-design-principles (дата звернення: 11.06.2026)

36. Логування (Logging) та аудит у програмних системах: кращі практики. *Highload.today*. URL: https://highload.today/uk/logging-audit-systems-python/ (дата звернення: 11.06.2026)

37. Моделювання загроз інформаційної безпеки для фінансових установ. *Cisco Networking Academy*. URL: https://netacad.com.ua/threat-modeling-finance/ (дата звернення: 11.06.2026)

38. PEPs (Politically Exposed Persons) та їх роль у фінансовому моніторингу. *Ліга:Закон*. URL: https://biz.ligazakon.net/pep-financial-monitoring/ (дата звернення: 11.06.2026)

39. UML: Діаграма компонентів (Component Diagram) у проектуванні систем. *SoftServe IT Academy*. URL: https://career.softserveinc.com/uk-ua/learning-materials/uml-component-diagram (дата звернення: 11.06.2026)

40. Особливості розробки кросплатформних застосунків на Python з використанням фреймворків. *DOU.ua*. URL: https://dou.ua/forums/topic/python-crossplatform-gui/ (дата звернення: 11.06.2026)

ДОДАТКИ

Додаток А

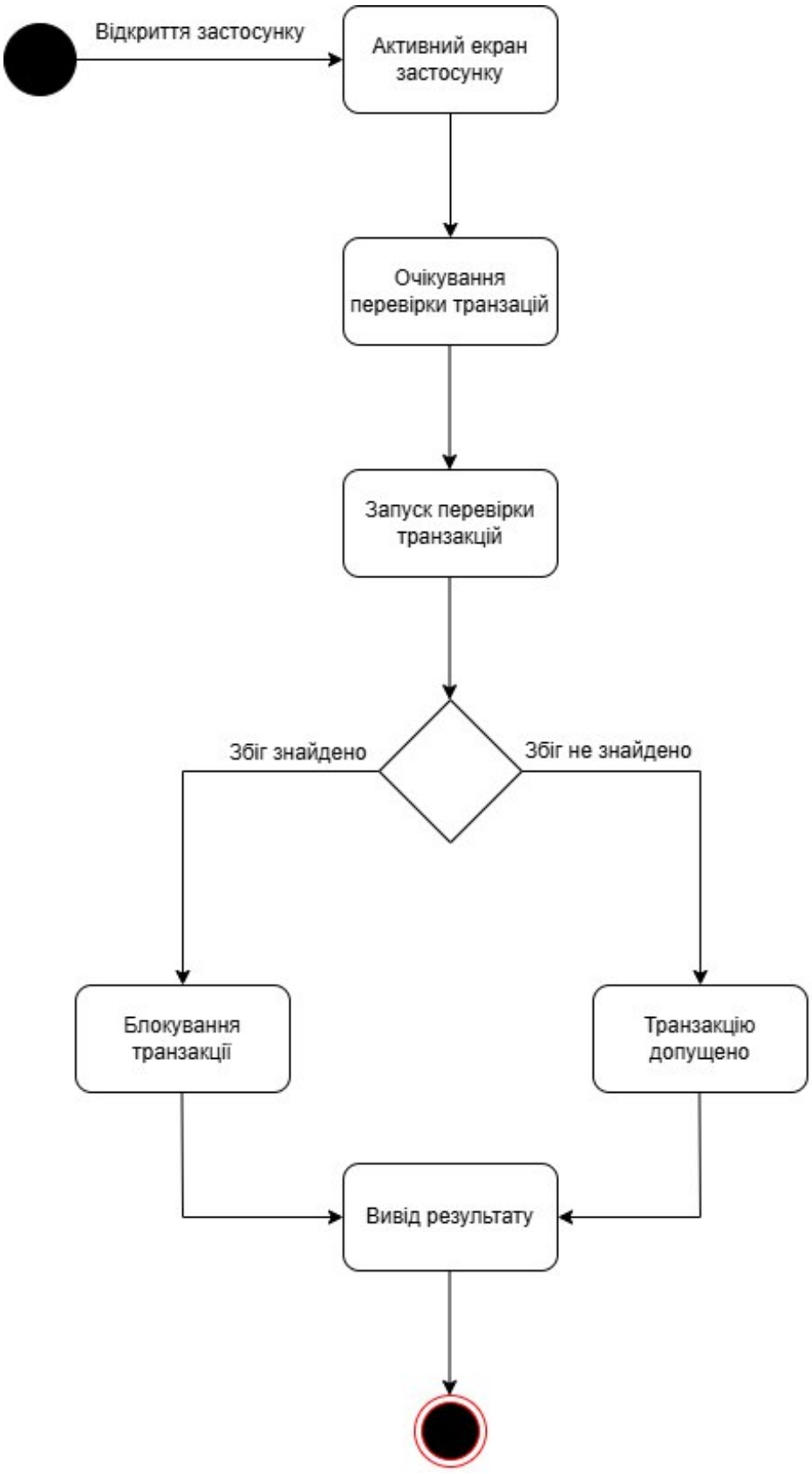

Рисунок А1 – Діаграма станів

Додаток Б

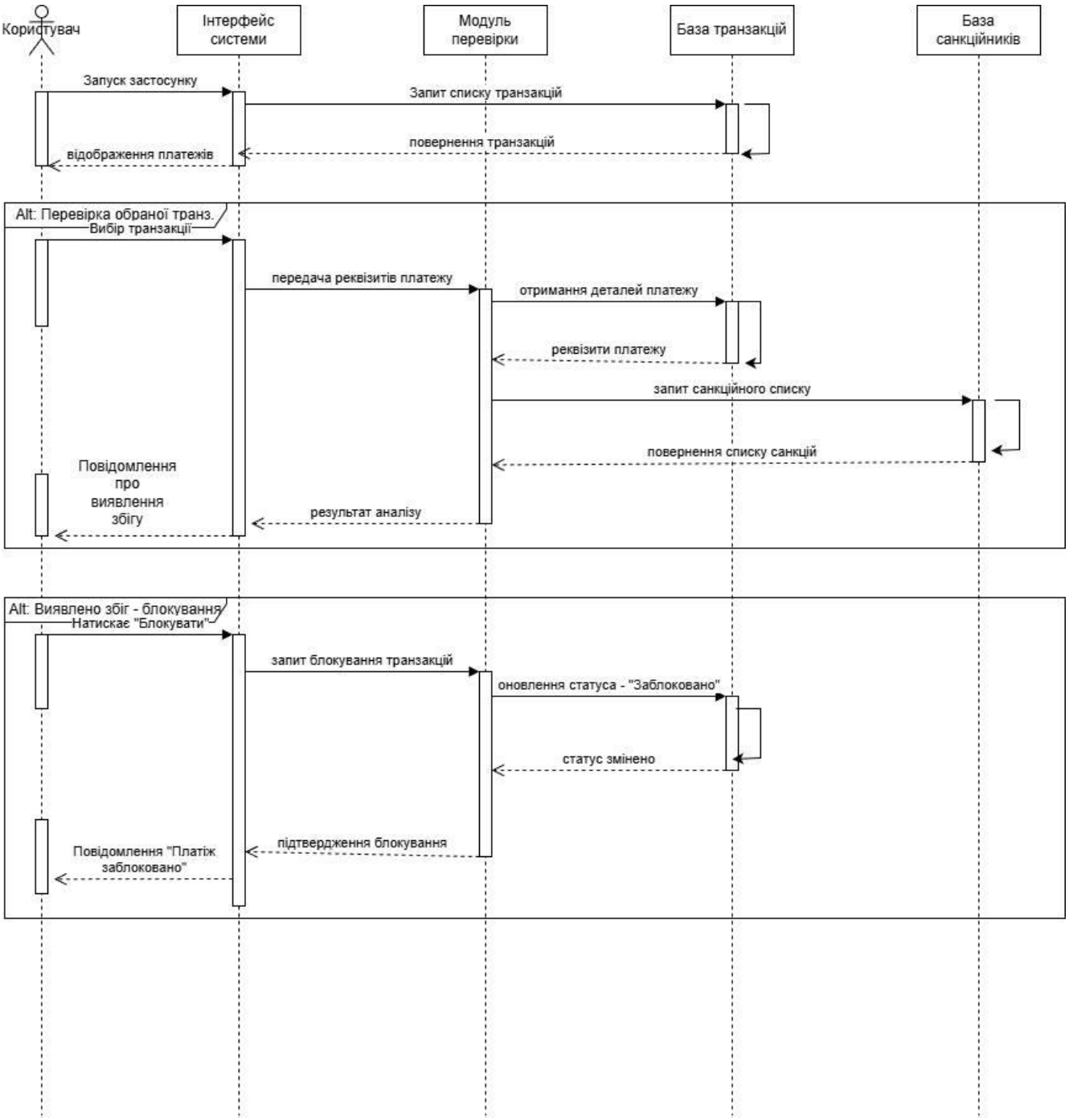

Рисунок Б1- Діаграма послідовності

Додаток В

```python
def fuzzy_search(self, name_to_check, cursor):
    cursor.execute("SELECT * FROM sanctions_list")
    all_sanctions = cursor.fetchall()
    best_match = None
    highest_ratio = 0.0

    for person in all_sanctions:
        sanction_name = person[2]
        if not sanction_name: continue
        # Обчислення метрики подібності
        ratio = difflib.SequenceMatcher(None, name_to_check.lower(), sanction_name.lower()).ratio()
        if ratio > highest_ratio:
            highest_ratio = ratio
            best_match = person

    # Перевірка проходження порогу у 85%
    if highest_ratio > 0.85:
        return best_match, highest_ratio
    return None, 0.0
```

Рисунок В1 - Фрагмент коду, що реалізує логіку порівняння та перевірку порогового значення $a = 85\%$