

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**

«До захисту допущено»

Т.в.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки» \_\_\_\_\_,

освітньо-наукової програми «Інформаційні технології проектування» \_\_\_\_\_

на тему: Інформаційна технологія підтримки діяльності операторів централізованих пультів спостереження за об'єктами \_\_\_\_\_

Здобувача (ки) групи ІТм-11н Рудик Віталій Віталійович \_\_\_\_\_  
(шифр групи) (прізвище, ім'я, по батькові)

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

\_\_\_\_\_ (підпис)

Віталій РУДИК  
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник \_\_\_\_\_ д.т.н., проф., Євгеній ЛАВРОВ \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

\_\_\_\_\_ (підпис)

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-наукова програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

В.о. зав. кафедри ІТ

\_\_\_\_\_ С. М. Ващенко

«\_\_\_» \_\_\_\_\_ 2023 р.

## **ЗАВДАННЯ**

на кваліфікаційну роботу магістра студентіві

Рудик Віталій Віталійович

(прізвище, ім'я, по батькові)

**1** Тема проекту *Інформаційна технологія підтримки діяльності операторів централізованих пультів спостереження за об'єктами*

затверджена наказом по університету від «05» травня 2023 р. № 0465-VI

**2** Термін здачі студентом закінченого проекту «\_\_\_» \_\_\_\_\_ 2023 р.

**3** Вхідні дані до проекту *Літературні джерела з питань ергономіки людино-машинних систем, інструкції, щодо діяльності бригад по усунення інцидентів, навчальна матриця для класифікації рівня загрози інцидентів.*

**4** Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *Аналіз предметної області; Постановка задачі та аналіз методів дослідження; Розробка інформаційної технології підтримки діяльності операторів централізованих пультів спостереження за об'єктами.*

**5** Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) *Об'єкт, предмет, мета та гіпотеза; Актуальність і передумови досліджень; Вибір методів реалізації; Дерево рішень для класифікації рівня загрози інцидентів; Функціональна мережа як засіб опису діяльності бригад; Принцип редукації функціональної мережі; Метод аналізу ієрархій для вирішення багатокритеріальної задачі розподілу інцидентів; Функціональні вимоги;*

Програмна реалізація інформаційної технології; Висновки; Апробація результатів роботи.

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання \_\_\_\_\_.

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	20.03.2023-22.03.2023	
2	Визначення вимог	23.03.2023-24.03.2023	
3	Аналіз і вибір методів реалізації	24.03.2023-25.03.2023	
4	Оцінка алгоритму діяльності бригад	25.03.2023-31.03.2023	
5	Реалізація класифікатора для визначення рівня загрози інциденту	01.04.2023-05.04.2023	
6	Реалізація бази даних	06.04.2023-12.04.2023	
7	Реалізація інформаційної технології з усіма підсистемами	13.04.2023-10.05.2023	
8	Тестування інформаційної технології	11.05.2023-14.05.2023	
9	Завершення проекту	14.05.2023-17.05.2023	

Магістрант \_\_\_\_\_

Рудик В.В.

Керівник роботи \_\_\_\_\_

д.т.н., проф., Лавров Є. А.

## РЕФЕРАТ

Тема роботи: «Інформаційна технологія підтримки діяльності операторів централізованих пультів спостереження за об'єктами».

Мета роботи: Розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами.

В роботі проаналізовано предметну область, розглянуто останні дослідження і публікації, а також програмні продукти-аналоги, визначено задачі та цілі розробки і обрано методи й засоби реалізації. Виконано реалізацію: бази даних, підсистеми визначення рівня загрози заявки, підсистеми аналізу і прогнозу безпомилковості реалізації алгоритму діяльності по усуненню небезпеки, підсистеми аналізу можливості виїзду і часу транспортування бригад, підсистеми визначення оптимального варіанту розподілу інцидентів, підсистеми обліку і документування інцидентів. Проведена інтеграція всіх підсистем в єдину інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами й тестування кінцевого продукту.

Практичне значення полягає в тому, що використання інформаційної технології підтримки діяльності оператора пульта спостереження за складними об'єктами забезпечує швидкий моніторинг інцидентів, місця знаходження бригад, вибір оптимальної бригади в залежності від обраних критеріїв, облік й керування робочим процесом при вирішенні інциденту.

Проект містить 127 сторінки, 5 таблиць, 45 рисунків, список літератури - 42 найменування, 6 додатків.

Ключові слова: ССППР, автоматизація, моделювання; оператор; математичні моделі; функціональна мережа; алгоритм; надійність; ймовірність; багатокритеріальні задачі; дерево рішень; метод аналізу ієрархій.

# ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Централізовані пульти спостереження за складними об'єктами.....	11
1.2 Інформатизація централізованих пультів спостереження за складними об'єктами в рамках концепції «Смарт-сіті».....	14
1.3 Аналіз досліджень і програмного забезпечення з проблематики спостереження та забезпечення безпеки складних об'єктів.....	16
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	23
2.1 Мета та задачі дослідження.....	23
2.2 Аналіз моделей для побудови класифікації рівня загрози заявки.....	24
2.3 Моделі та методи вирішення задач багатокритеріальної оптимізації.....	26
2.4 Математична модель методу аналізу ієрархій.....	30
2.5 Методи формалізації алгоритму діяльності бригади.....	33
2.6 Вибір засобів реалізації.....	35
3 МОДЕЛЮВАННЯ Й ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	37
3.1 Структурно-функціональне проектування.....	37
3.2 Моделювання варіантів використання.....	39
3.3 Проектування моделі бази даних.....	41
4 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПІДТРИМКИ ДІЯЛЬНОСТІ ОПЕРАТОРІВ ЦЕНТРАЛІЗОВАНИХ ПУЛЬТІВ СПОСТЕРЕЖЕННЯ ЗА ОБ'ЄКТАМИ.....	43
4.1 Реалізація користувальницького інтерфейсу.....	43
4.2 Реалізація підсистеми визначення рівня загрози.....	46
4.3 Реалізація підсистеми оцінки часу транспортування бригад на місце події.....	49

4.4 Реалізація підсистеми аналізу і прогнозу безпомилковості реалізації алгоритму діяльності бригад.....	51
4.5 Реалізація підсистеми вибору оптимального варіанту розподілу інцидентів.....	58
4.6 Реалізація підсистеми обліку і документування інцидентів безпеки .....	63
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТОК А. Планування робіт .....	73
ДОДАТОК Б. Перелік типових функціональних одиниць .....	79
ДОДАТОК В. Перелік типових функціональних структур.....	81
ДОДАТОК Г. Фрагменти кодів реалізації .....	84
ДОДАТОК Д. Фрагменти кодів реалізації.....	115
ДОДАТОК Е. Акт впровадження .....	127

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CMS – Централізований пульт спостереження (Central Monitoring System)

GPS – Система глобального позиціонування (Global Positioning System)

IoT – Інтернет речей (Internet of Things)

AI – Штучний інтелект (Artificial Intelligence)

API – Інтерфейс програмування застосунків (Application Programming Interface)

IDEF0 – Методологія моделювання функцій (Integrated Definition for Function Modeling)

UML – Уніфікована мова моделювання (Unified Modeling Language)

MAI – Метод аналізу ієрархій

СППР – Система підтримки прийняття рішень

ФМ – Функціональна мережа

## ВСТУП

**Актуальність.** Дослідження виконується на замовлення підприємства ЦТО УПО в Сумській області. Останнім часом підвищились рівні небезпеки для різноманітних об'єктів інфраструктури населених пунктів та приватних господарств. В зв'язку з такими проблемами створюються спеціальні підрозділи управління безпекою. Такі підрозділи мають бути обладнані системами підтримки прийняття рішень. Оператори спеціальних підрозділів охорони складних об'єктів керують процесами усунення проблемних ситуацій та управляють діяльністю оперативних мобільних бригад. З метою забезпечення діяльності операторів розробляють спеціальні інформаційні системи, які повинні бути обладнані підсистемами підтримки прийняття рішень, побудованими за принципами варіантного аналізу і оптимізації діяльності охоронців. Не дивлячись на велику увагу практиків і науковців до проблематики ефективності централізованих пультів спостереження (в т.ч. в рамках концепції “розумне місто”), задача управління оперативними мобільними бригадами на основі об'єктивних кількісних показників вирішена не до кінця).

**Об'єкт дослідження.** Діяльність людини-оператора централізованого пульта спостереження за складними об'єктами.

**Предмет дослідження.** Підтримка прийняття рішень людини-оператора централізованого пульта спостереження за складними об'єктами при призначенні бригад для усунення інцидентів.

**Наукова новизна.** На відміну від існуючих моделей інформаційного забезпечення діяльності операторів пульта спостереження, орієнтованих на інтуїтивні методи прийняття рішень, розроблені моделі використовують формальний апарат опису діяльності по усуненню інцидентів на основі методології функціональних мереж та апарат ієрархічної декомпозиції



проблеми, який дозволяє приймати рішення на основі багатокритеріального оцінювання альтернативних сценаріїв усунення інцидентів.

**Гіпотеза дослідження.** Якщо використати функціональну мережу в якості моделі опису усунення інциденту та методологію ієрархічної декомпозиції проблеми, можна забезпечити прийняття оптимальних рішень з врахуванням багатокритеріальності проблеми усунення інцидентів безпеки.

**Мета.** Розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами.

**Основні задачі.** Розглянути предметну область, проаналізувати останні дослідження на схожу тематику й продукти - аналоги, визначити методи, засоби, технології й вимоги розроблюваного продукту, обрати моделі для реалізації підтримки діяльності оператора, спроектувати й розробити базу даних для інформаційної технології, реалізувати підсистему визначення рівня загрози заявки, реалізувати підсистему аналізу і прогнозу безпомилковості реалізації алгоритму діяльності по усуненню небезпеки, реалізувати підсистему аналізу можливості виїзду і часу транспортування бригад, реалізувати підсистему визначення оптимального варіанту розподілу інцидентів, реалізувати підсистему обліку і документування інцидентів, розробити й протестувати інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами.

**Практичне значення.** Використання інформаційної технології підтримки діяльності оператора пульта спостереження за складними об'єктами забезпечує швидкий моніторинг інцидентів, місця знаходження бригад, вибір оптимальної бригади в залежності від обраних критеріїв, облік й керування робочим процесом при вирішенні інциденту.

**Апробація результатів:** Результати доповідались на 2 наукових конференціях:

Науково технічна конференція «Інформатика Математика Автоматика», м. Суми: СумДУ, 19-23 квітня 2021;

Науково технічна конференція «Інформатика Математика Автоматика», м. Суми: СумДУ, 24-28 квітня 2023.

**Публікації.** За матеріалами досліджень опубліковано 5 наукових робіт [1, 2, 3, 4, 5] (копії наведено в додатку Д).

**Використання результатів та впровадження.** Дослідження проводились на замовлення підприємства “Центр технічної охорони. Управління поліції охорони” (ЦТО УПО) в Сумській області протягом 2019-2023 рр. Магістерська робота є продовженням робіт, представлених в рамках бакалаврської роботи [6], матеріали якої частково використані при реалізації підсистеми аналізу і прогнозу безпомилковості алгоритму діяльності бригад по усуненню інцидентів. Результати послідовно проходять апробацію та впроваджуються на підприємстві ЦТО УПО в Сумській області. На жаль, через посилення режимних обмежень, підприємство не має дозволу на формування актів впровадження. В зв’язку з цим надається акт впровадження на частину роботи (підсистема аналізу і прогнозу безпомилковості алгоритму діяльності бригад по усуненню інцидентів), наданий підприємством в 2021р. (наведено в додатку Е).

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Централізовані пульти спостереження за складними об'єктами

Централізований пульт спостереження (англ. Central Monitoring System, CMS) - це система, яка використовується для моніторингу та контролю різних видів об'єктів (наприклад, промислових підприємств, транспортних систем, банків, мереж інтернет-комерції тощо) за допомогою відеокамер, сенсорів, датчиків, GPS-модулів та іншого обладнання. CMS може забезпечувати різноманітні функції, такі як відображення відеопотоків з камер, збір та аналіз даних з датчиків, моніторинг роботи обладнання та систем, дистанційне управління об'єктами, аналіз поведінки користувачів та ризиків безпеки, інтеграцію з іншими системами тощо. Такі пульти зазвичай використовуються в охоронних компаніях, правоохоронних органах, банківських установах, транспортних компаніях, промислових підприємствах та інших організаціях, де важливо забезпечити високий рівень безпеки, контролю та ефективності роботи [7].

Важливою ланкою в централізованих пультах спостереження є оператор. Він зазвичай відповідає за відслідковування, аналіз та реагування на різноманітні типи подій та інцидентів, які стосуються об'єктів, що перебувають під наглядом [8]. Оператор може працювати з різними типами систем, включаючи системи відеоспостереження, системи сигналізації, системи контролю доступу та інші. Робоче місце оператора (рис. 1.1) централізованого пульта спостереження може виглядати по-різному залежно від конкретних вимог та типу спостережуваних об'єктів. Основні елементи його робочого місця включають в себе:

Монітори: оператор може працювати з декількома моніторами, на яких відображаються відеопотоки з камер, статистичні дані, картографічні дані, сповіщення тощо;

– Клавіатура, миша та інші периферійні пристрої: оператор взаємодіє з різними системами та програмами за допомогою клавіатури, миші та інших периферійних пристроїв;

– Система відеоспостереження: система може включати в себе відеокамери, які дозволяють відстежувати події на об'єктах, а також забезпечувати віддалений доступ до зображення з цих камер;

– Система сигналізації: система може забезпечувати виявлення та сповіщення оператора про різні типи подій, такі як відкриті двері, злам системи безпеки, проникнення на територію об'єкта та інші;

– Система контролю доступу: система може забезпечувати контроль доступу до об'єктів, а також відслідковувати та аналізувати поведінку людей на території об'єкта;

– Система збору та аналізу даних: система може забезпечувати збір та аналіз даних з різних джерел, таких як датчики, GPS-модулі, системи відеоспостереження, інші.



Рисунок 1.1 – Робоче місце оператора

Централізовані пульти спостереження за об'єктами стали важливим інструментом для багатьох організацій і установ, які мають складні об'єкти або мережі об'єктів, що потребують нагляду та безпеки. Вони допомагають забезпечити ефективний моніторинг, відслідковування та реагування на події в режимі реального часу. Одним із викликів, що вирішуються цими пультами, є забезпечення безпеки та захисту об'єктів. Зокрема, це може стосуватися забезпечення безпеки людей та майна, зниження ризику крадіжок, зламів та інших небажаних подій. Крім того, такі пульти допомагають забезпечувати ефективний моніторинг і контроль за роботою обладнання, що дозволяє знижувати ймовірність аварій та непередбачених ситуацій. Ще одним важливим викликом, що допомагають вирішувати централізовані пульти, є підвищення ефективності роботи організацій. Завдяки автоматизації багатьох процесів, спрощенню доступу до інформації та підвищенню оперативності взаємодії між різними підрозділами, пульт дозволяє розподіляти робочі завдання та керувати ресурсами ефективніше, що в свою чергу забезпечує зростання продуктивності і прибутковості організації [9].

Одним з можливих напрямів розвитку централізованих пультів спостереження є їх інтеграція з системами штучного інтелекту та аналізу даних. Застосування штучного інтелекту для аналізу відеопотоку та інших даних, що надходять на пульт спостереження, дозволить забезпечити більш ефективну роботу операторів та зменшити кількість помилок [10]. Іншим напрямом є використання дронів для забезпечення спостереження за об'єктами. Дрони можуть бути оснащені камерами та іншими датчиками для отримання детальної інформації про об'єкт та його навколишнє середовище. Інформація, отримана з дронів, може бути передана на пульт спостереження для подальшого аналізу та обробки [11]. Таким чином, централізовані пульти спостереження за об'єктами є невід'ємною частиною сучасного світу та виконують важливу роль в забезпеченні безпеки та ефективного управління різними об'єктами. Розвиток

технологій та їх інтеграція з цими системами дозволить забезпечити ще більш високий рівень безпеки та ефективності.

## **1.2 Інформатизація централізованих пультів спостереження за складними об'єктами в рамках концепції «Смарт-сіті»**

Смарт-сіті (Smart City) - це концепція розвитку міст, яка передбачає використання різноманітних інформаційно-комунікаційних технологій та інноваційних рішень з метою покращення якості життя мешканців та оптимізації управління містом. У такому місті використовуються різноманітні датчики, системи збору даних та аналізу, що дозволяє в режимі реального часу моніторити стан інфраструктури, доріг, транспорту, енергосистем, екології та інших аспектів функціонування міста. Така інтеграція технологій дозволяє підвищити ефективність та якість послуг, сприяє розвитку бізнесу та туризму, покращує безпеку міста та рівень задоволеності мешканців життям в ньому [12].

Інформаційно-комунікаційні технології, що використовуються в смарт-сіті, дозволяють збирати та аналізувати великі масиви даних про різні сфери життєдіяльності міста - від стану доріг та комунальних послуг до поведінки мешканців та екологічних показників. На основі цих даних розробляються різноманітні сервіси та ініціативи, які спрямовані на поліпшення життя мешканців та оптимізацію роботи міських служб. До прикладів реалізації концепції смарт-сіті можна віднести встановлення датчиків на вуличне освітлення, що дозволяє зменшити витрати на енергопостачання, використання системи автоматичної оплати громадського транспорту, що забезпечує зручність та швидкість проїзду містом, впровадження сервісу електронного урядування, що спрощує процедуру взаємодії [13].

Концепція смарт-сіті охоплює різні аспекти життя міста, включаючи транспорт, енергетику, водопостачання, відходи, безпеку, а також управління даними та інформацією. Головною метою є забезпечення життєздатності міста та підвищення якості життя мешканців.

Смарт-сіті зазвичай базуються на використанні новітніх технологій, таких як Інтернет речей (IoT), штучний інтелект (AI), аналітика даних, блокчейн та інші. Застосування цих технологій дозволяє збирати, аналізувати та обробляти великі обсяги даних з різних джерел, що дозволяє приймати обґрунтовані рішення та підвищувати ефективність управління містом [14].

Смарт-сіті передбачає інтеграцію різноманітних інформаційних технологій та інфраструктур для кращого управління містами та забезпечення більш ефективних послуг для мешканців. Одним із напрямків розвитку смарт-сіті є використання систем централізованих пультів спостереження за складними об'єктами, які дозволяють контролювати роботу різноманітних систем в режимі реального часу. Ці системи стали надзвичайно актуальними в останні роки зі зростанням обсягів даних та підвищенням складності об'єктів, які потрібно контролювати. Це особливо стосується об'єктів критичної інфраструктури, таких як електростанції, водопроводи, метрополітени та інші. У разі виникнення аварій чи порушень у роботі цих систем, ефективна реакція операторів пульта спостереження може врятувати життя та майно тисяч людей [15].

Зв'язок між смарт-сіті та централізованими пультами спостереження полягає у тому, що в сучасних містах стає все більше складними об'єкти та інфраструктура, що вимагає ефективної системи моніторингу та управління. Саме тут інформаційні технології та системи підтримки прийняття рішень, що забезпечують роботу централізованих пультів спостереження, можуть допомогти досягти ефективної організації та управління містом. Таким чином, розробка та впровадження інформаційної технології підтримки діяльності оператора пульта спостереження є актуальною темою у контексті побудови смарт-сіті та покращення якості життя мешканців міст.

### **1.3 Аналіз досліджень і програмного забезпечення з проблематики спостереження та забезпечення безпеки складних об'єктів**

Централізовані пульти спостереження за складними об'єктами зазвичай включають в себе різні типи систем моніторингу, які забезпечують збір інформації з датчиків і камер спостереження, а також системи аналізу даних та прийняття рішень. До таких систем відносяться:

– Системи відеоспостереження: ці системи забезпечують високоякісний збір відеоданих з камер спостереження. Вони можуть використовуватися для моніторингу будь-якого типу об'єктів, включаючи важливі промислові та комерційні об'єкти. В роботі “Розпізнавання аномальної поведінки для інтелектуальних систем відеоспостереження” [16] автори проаналізували та порівняли різні підходи, що використовуються для виявлення та класифікації незвичайної поведінки, зокрема методи на основі машинного навчання та глибокого навчання. Огляд включає опис технологій, їхніх переваг та недоліків, а також можливостей використання для вирішення різних завдань в інтелектуальних системах відеоспостереження.

– Системи управління доступом: ці системи забезпечують контроль доступу до об'єктів та зон певних приміщень. Вони можуть використовуватися для забезпечення безпеки на території об'єктів і зменшення ризику крадіжок та інших злочинів. Таку систему розглядали в роботі "Інтегрована система контролю доступу та налаштування освітлення для розумних будівель" [17]. Автори описують розробку та імплементацію системи управління доступом та освітленням для розумних будівель, пропонують рішення, яке поєднує в собі можливості контролю доступу до будівлі та управління освітленням. У статті детально описано архітектуру системи, що складається з різних компонентів, таких як читачі карток, контролери доступу, датчики руху та система керування



освітленням. Автори також розглядають питання безпеки та приватності при роботі з такою системою.

– Системи виявлення вторгнень: ці системи забезпечують виявлення підозрілих дій на об'єкті і сповіщення оператора пульта спостереження про можливу загрозу безпеці. Вони можуть включати в себе різні технології, такі як датчики руху, датчики вібрації та системи звуку. Стаття "Система виявлення вторгнень на основі IoT за допомогою датчика PIR" [18] описує розробку системи виявлення вторгнень на основі технологій Інтернету речей (IoT) та датчика руху. Автори статті, Khirod Chandra Sahoo та Umesh Chandra Pati, пропонують рішення, яке дозволяє виявляти незаконний доступ до приміщень за допомогою датчика руху PIR (Passive Infrared). Дані з датчика передаються на базову станцію, яка підключена до Інтернету та надсилає повідомлення на мобільний телефон власника будівлі у разі виявлення незвичайного руху. Також детально описано архітектуру системи, яка включає в себе мікроконтролер ESP8266, датчик PIR, Wi-Fi модуль та базову станцію. Також описано процес налаштування та тестування системи. У підсумку, стаття демонструє потенційні можливості використання технологій IoT та датчиків руху для створення ефективних систем виявлення вторгнень у приміщення.

– Системи GPS-моніторингу: ці системи забезпечують моніторинг геопозиції транспортних засобів та інших рухомих об'єктів на об'єкті. Це може бути корисним для відстеження маршруту перевезення цінних вантажів, наприклад. В роботі "Система моніторингу місцезнаходження на основі GPS із можливостями геозонування" [19] розглядається проблема відстеження місцезнаходження мобільних об'єктів, таких як автомобілі, вантажівки, мотоцикли тощо, які зазвичай використовуються в транспортних компаніях та логістичних центрах. Система дозволяє відстежувати місцезнаходження транспортного засобу з використанням GPS-координат, що передаються на сервер у режимі реального часу. Додатково, система має можливість

налаштування гео-огорожі, що дозволяє встановлювати обмеження на маршрут руху транспорту та отримувати сповіщення у разі порушення цих обмежень.

– Системи розпізнавання облич: ці системи забезпечують автоматичне розпізнавання облич та ідентифікацію осіб на об'єкті. Це може бути корисним для контролю доступу та ідентифікації підозрілих осіб на території об'єкту. В роботі “Розробка вбудованої системи домашньої безпеки на основі розпізнавання облич” [20] висвітлюється проблема забезпечення безпеки в домашніх умовах та розробляється система, яка використовує технологію розпізнавання обличчя для ідентифікації користувачів та контролю доступу до будинку. Система містить вбудований модуль з обробки зображень та нейронну мережу для розпізнавання обличчя. У статті описано архітектуру системи та процес її розробки, а також детально розглядаються алгоритми обробки зображень та навчання нейронної мережі. Додатково, автори проводять експерименти для оцінки ефективності системи та порівнюють результати з іншими методами контролю доступу. У підсумку, демонструється можливість використання технології розпізнавання обличчя для створення ефективної системи домашньої безпеки.

Якщо розглядати безпосередньо програми-аналоги, які мають схожий функціонал і можуть допомогти вирішити задачі поставлені в роботі, то варто звернути увагу на найбільш популярні сервіси, що надають можливість моніторити об'єкти, бригади на мапі та керувати ними, такі як Geotab, Webfleet Solutions, Fleetio, Wialon.

Geotab - це інтегрована платформа для управління транспортом (рис 1.2), яка дозволяє моніторити рух транспортних засобів на мапі, використовуючи GPS-модулі, та збирати дані про їх рух та стан. Geotab також має функції аналізу даних, що дозволяє покращувати продуктивність та ефективність бригад [21].

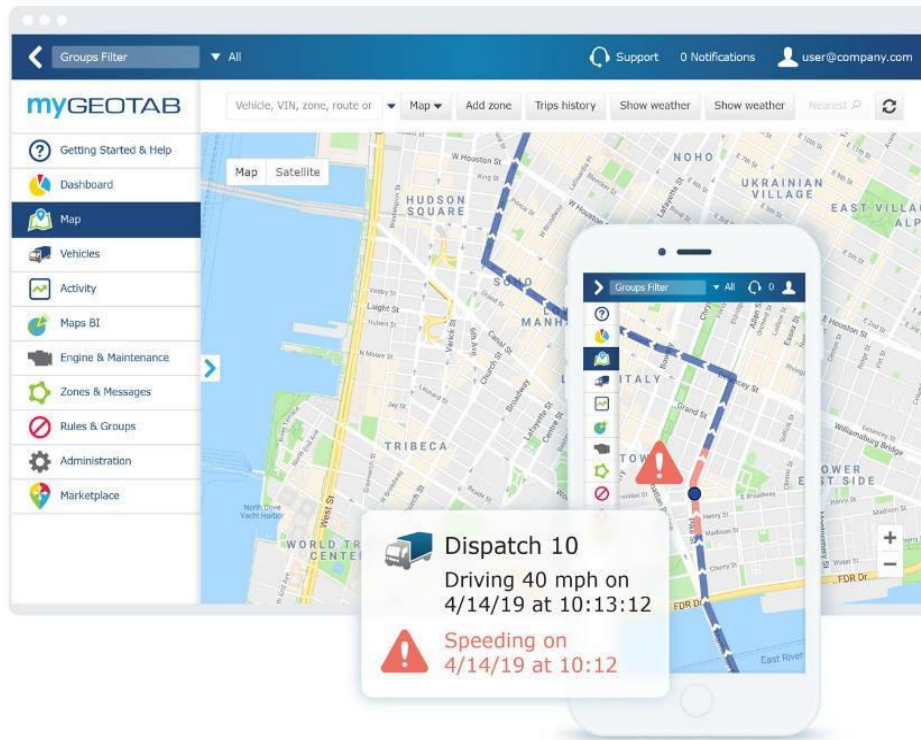


Рисунок 1.2 – Інтерфейс програми Geotab [21]

Webfleet Solutions - це постачальник технологій GPS-відстеження, управління транспортними флотами та телематики (рис 1.3). Компанія надає широкий спектр послуг та рішень для управління транспортними флотами різного розміру та складності.

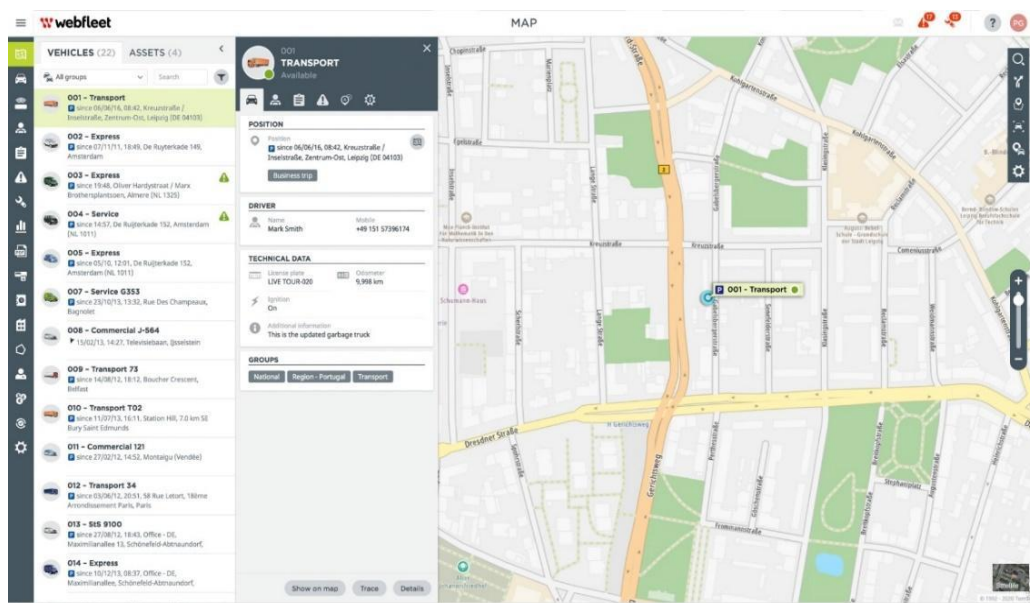


Рисунок 1.3 – Інтерфейс програми Webfleet Solutions [22]

Webfleet Solutions має функції відстеження транспорту і його управління, планування маршрутів, аналізу даних про транспорт, моніторингу й оптимізації стилю водіння [22].

Fleetio - це програмне забезпечення для управління транспортними бригадами (рис 1.4). Воно надає рішення для моніторингу транспортних засобів, планування та ремонту, управління витратами, аналіз даних про транспорти, відстеження стану авто, створення бази даних транспортів та багато іншого [23].

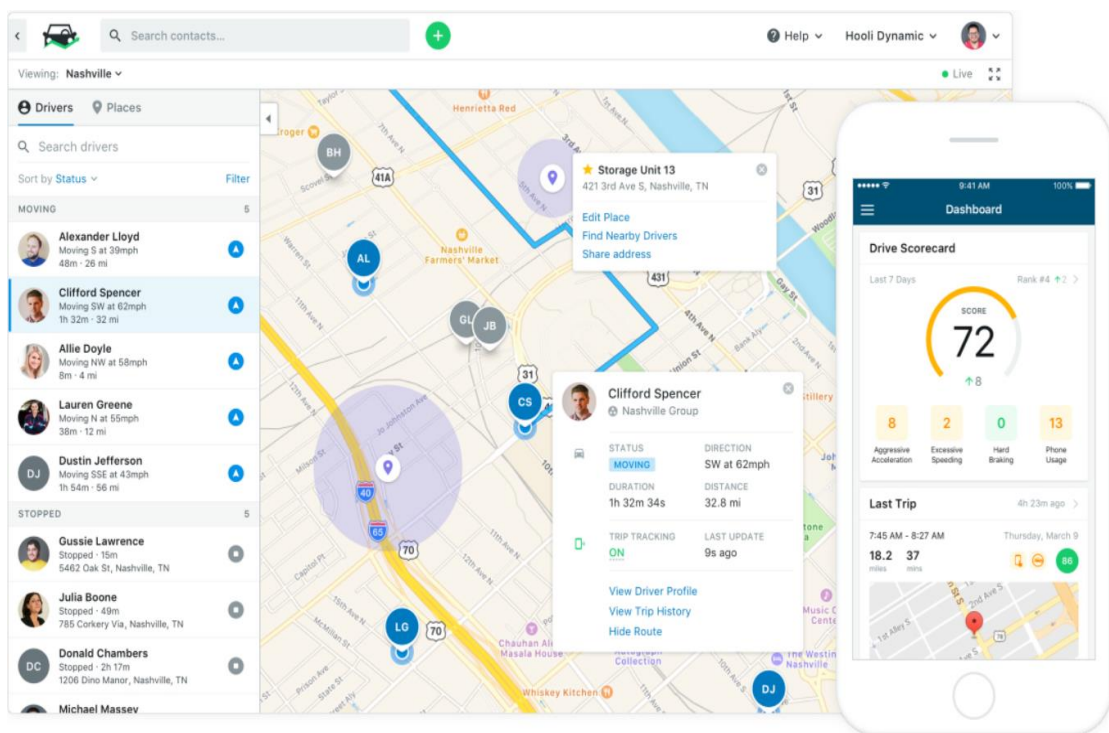


Рисунок 1.4 – Інтерфейс програми Fleetio [23]

Wialon - це програмне забезпечення для моніторингу транспортних засобів і управління транспортними флотами (рис 1.5). Воно надає можливості для відстеження місцезнаходження транспорту, аналізу маршруту і швидкості руху транспорту, контролю стану авто, споживання палива та інших параметрів [24].

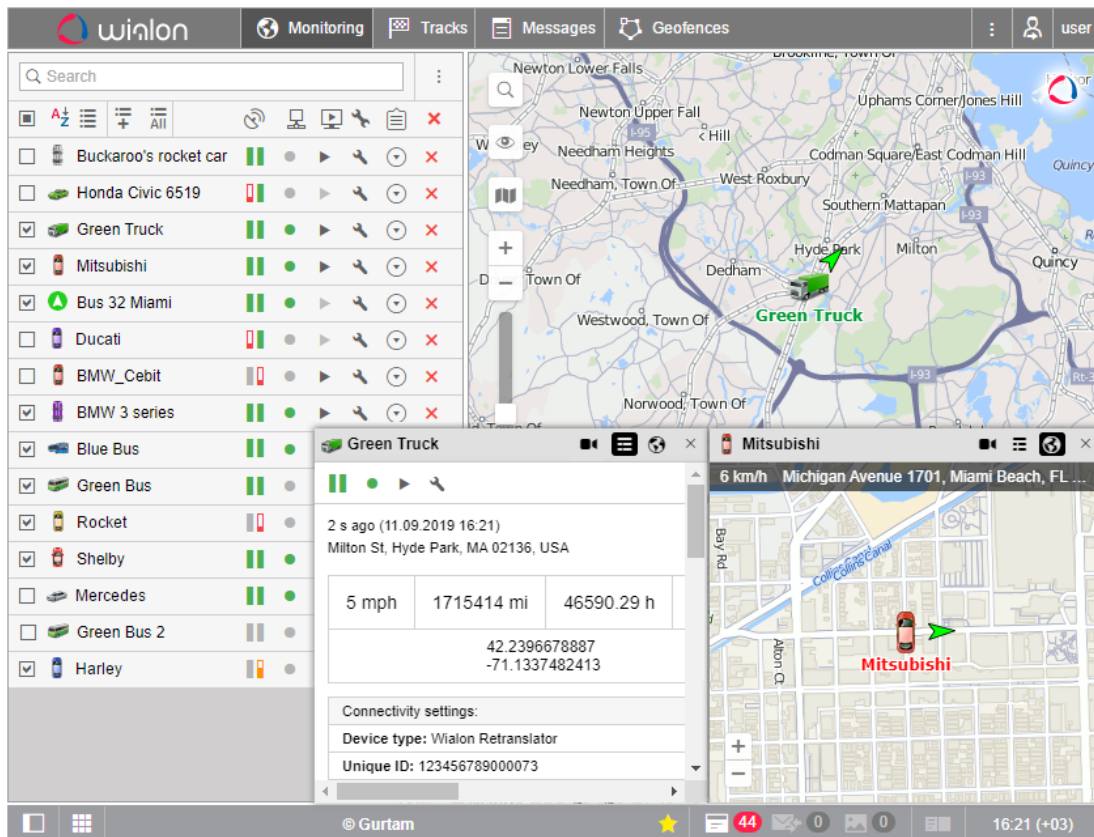


Рисунок 1.5 – Інтерфейс програми Wialon [24]

Переваги й недоліки кожної програми у вирішенні задач дослідження наведено у таблиці 1.1.

В цілому розглянуті програми мають схожий функціонал і виконують одні й ті ж задачі, але основний їх недолік в тому що вони створені для бізнес задач, і ніяк не підходять як системи для використання в централізованому пульті спостереження за об'єктами. В них не передбачена, або обмежена можливість інтеграції з іншими системами, що робить неможливим впровадження системи підтримки прийняття рішень, та інших підсистем для покращення ефективності роботи оператора централізованого пульта спостереження за об'єктами.

Таблиця 1.1 – Порівняння програм-аналогів.

Програма	Переваги	Недоліки
Geotab	Висока точність визначення місцезнаходження, простий та зручний інтерфейс користувача, простота в налаштуванні	Обмежені можливості управління бригадами та маршрутами, висока вартість, відсутність можливості інтеграції з іншими системами
Webfleet Solutions	Збір даних про рух транспорту та виконання завдань, можливість налаштування сповіщень, оптимізація маршрутів	Висока вартість, обмежені можливості в налаштуванні, відсутність можливості інтеграції з іншими системами, незручний інтерфейс
Fleetio	Велика кількість звітів та аналітичних інструментів, можливість автоматичного сповіщення бригад, висока точність визначення місцезнаходження	Обмежений функціонал порівняно з іншими системами, - обмежена можливість інтеграції з іншими системами, відсутність можливості налаштування повідомлень
Wialon	Простий та зручний інтерфейс користувача, збір даних про рух транспорту та виконання завдань, дуже точний моніторинг стану авто	Не дуже точне визначення місцезнаходження в деяких місцях, обмежені можливості управління маршрутами, відсутність інтеграції з іншими системами

## 2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

### 2.1 Мета та задачі дослідження

В роботі була поставлена мета розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами. Вона має бути орієнтована на облік заявок і управління інцидентами в системах забезпечення безпеки складних об'єктів; моделювання варіантів призначення на реалізацію поточних інцидентів оперативних мобільних бригад і оптимізацію розподілу заявок між бригадами. Для досягнення мети необхідно виконати наступні задачі:

- обрати моделі для реалізації підтримки діяльності оператора;
- розробити підсистему аналізу і прогнозу безпомилковості реалізації алгоритму діяльності по усуненню небезпеки;
- спроектувати й розробити базу даних для інформаційної технології;
- розробити інтерфейс користувача інформаційної технології;
- реалізувати картографічні функції, що забезпечують відображення карт та об'єктів на ній;
- реалізувати інтерфейс перегляду, додавання, редагування об'єктів, власників, патрульних, бригад;
- реалізувати підсистему визначення рівня загрози заявки;
- реалізувати підсистему аналізу можливості виїзду і часу транспортування бригад;
- реалізувати підсистему визначення оптимального варіанту розподілу інцидентів;
- реалізувати підсистему обліку і документування інцидентів;
- протестувати реалізовану інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами.

Під час формулювання мети і задач було виконане планування робіт та створено календарний план які наведені у додатку А.

## **2.2 Аналіз моделей для побудови класифікації рівня загрози заявки**

Для реалізації класифікатора що лежить в основі підсистеми визначення рівня загрози заявки і враховує такі фактори як: категорія об'єкту де спрацювала сигналізація (від 1 до 3, в залежності від значущості об'єкта) кількість повторних спрацювань (кількість спрацювань сигналізації після початкового спрацювання), кількість зон спрацювання (кількість датчиків на об'єкті, що спрацював) та наявність об'єктів, що створюють шуми, можна використати модель Байєса, модель дерева рішень або нейронні мережі.

Модель Байєса базується на ймовірностях і використовує статистичні методи для розподілу вхідних даних на класи. У контексті класифікації, модель Байєса може бути використана для призначення категорії або класу новому елементу, основувшись на попередніх знаннях про властивості інших елементів у відповідних категоріях. Наприклад, модель Байєса може бути використана для класифікації електронного листа як спаму або не спаму, основувшись на його властивостях, таких як наявність певних слів або фраз у тексті. Однією з головних переваг моделі Байєса є її висока ефективність при роботі з великою кількістю признаков або властивостей, які можуть бути використані для класифікації. Також вона дозволяє враховувати нові дані при кожному новому класифікуванні, що дозволяє покращувати точність класифікатора з часом [25].

Хоч модель Байєса має багато переваг, вона також має свої недоліки, які потрібно враховувати при її використанні:



- Попередні знання: модель Байєса потребує попередніх знань про взаємозв'язок між признаками, що може бути складно або навіть неможливо отримати в деяких ситуаціях;

- Наявність припущень: модель Байєса заснована на деяких припущеннях, наприклад, що всі признаки незалежні між собою. Ці припущення можуть бути неприйнятними для деяких задач або даних, що може призвести до поганої точності класифікації;

- Проблема з рідкими даними: модель Байєса може мати проблему з рідкими даними, коли деякі можливі комбінації признаків не спостерігаються в навчальному наборі даних. Це може призвести до низької точності класифікації;

- Чутливість до викидів: модель Байєса може бути чутливою до викидів або ненормальних значень, що може призвести до поганої точності класифікації.

Модель дерев рішень - це одна з найпоширеніших моделей машинного навчання для задач класифікації, яка використовує деревоподібну структуру для прийняття рішень. Модель дерев рішень використовується в різних галузях, включаючи медицину, бізнес, банківську справу, науку про матеріали та багато інших. Вони легко інтерпретуються, що дозволяє зрозуміти, як модель виконує класифікацію та встановити, які атрибути найбільше впливають на результат. Класифікація в моделі дерев рішень зазвичай здійснюється дуже швидко, що дозволяє її використовувати в реальному часі [26]. Серед недоліків моделі можна виділити наступні:

- Схильність до перенавчання: модель дерев рішень може стати дуже складною та перенавчитися на навчальних даних, що може призвести до поганої точності на нових даних;

- Схильність до зміни даних: дерева рішень можуть бути дуже чутливі до незначних змін в даних, що може призвести до іншого результату класифікації;

- Проблема з балансом класів: Дерева рішень можуть мати проблеми з балансом класів, оскільки вони можуть створювати неправильні або неадекватні рішення для менш численних класів.

Нейронні мережі: нейронні мережі є однією з найпотужніших інструментів машинного навчання для створення класифікаторів, вони засновані на математичній моделі, що імітує роботу людського мозку. Нейронні мережі можуть виявляти складні зв'язки між даними, що робить їх ефективними для розв'язання задач класифікації, де дані можуть мати складну структуру. Також серед переваг є те, що вони можуть автоматично виконувати вибір та генерування ознак з даних, що робить їх ефективними для роботи з великими та складними даними [27]. Але вони мають й свої недоліки:

- Велика кількість параметрів: нейронні мережі можуть мати велику кількість параметрів, що може викликати проблеми з навчанням та перенавчанням;

- Високі вимоги до обчислювальних ресурсів: навчання нейронних мереж може вимагати великої кількості обчислювальних ресурсів, що може стати проблемою при використанні на великих наборах даних;

- Необхідність великої кількості даних: нейронні мережі можуть вимагати великої кількості даних для навчання.

Проаналізувавши всі особливості моделей що надають можливість створити класифікатор для визначення рівня загрози було обрано дерева рішень для реалізації підсистеми, адже кількість даних для навчання не є дуже великою, що унеможлиблює використання нейронних мереж, також потрібна висока точність роботи без припущень, чого не можуть надати моделі Байєса. Також важливим фактором у виборі моделі є можливість швидкої класифікації в режимі реального часу, що надає модель дерева рішень.

## **2.3 Моделі та методи вирішення задач багатокритеріальної оптимізації**

Багатокритеріальна оптимізація - це галузь оптимізації, в якій метою є знайти набір оптимальних рішень з урахуванням декількох конфлікуючих

критеріїв. Існує багато моделей та методів вирішення задач багатокритеріальної оптимізації:

– Методи вагової суми: ці методи вирішують задачу шляхом вагового присвоєння кожному критерію ваги і обчислення взваженої суми кожного варіанту рішень. Однак ці методи не враховують взаємодії між критеріями.

– Метод аналізу ієрархій: цей метод ґрунтується на визначенні ієрархії критеріїв та їх вагомості. Він дозволяє враховувати взаємодії між критеріями.

– Метод змішаної цільової функції: цей метод комбінує кілька критеріїв в одну цільову функцію. Ця функція зазвичай включає вагові коефіцієнти для кожного критерію.

– Метод еволюційної оптимізації: цей метод заснований на алгоритмах генетичного пошуку, еволюційних стратегій та інших методах. Він може використовуватись для розв'язання багатокритеріальних задач, що мають складні, нелінійні цільові функції та обмеження.

– Метод парето-оптимальності: цей метод базується на теорії Парето-оптимальності, яка стверджує, що розв'язок називається Парето-оптимальним, якщо неможливо поліпшити значення одного критерію, не погіршуючи значення інших критеріїв. Цей метод дозволяє знайти набір Парето-оптимальних рішень.

– Метод адаптивного вибору: цей метод використовується для покращення методів вагової суми та змішаної цільової функції. Він дозволяє вибирати вагові коефіцієнти таким чином, щоб досягти найкращих результатів.

– Метод декомпозиції: цей метод розбиває багатокритеріальну задачу на кілька підзадач, кожна з яких має один критерій. Підзадачі розв'язуються окремо, а потім об'єднуються, щоб отримати кінцевий набір рішень.

– Метод згортки критеріїв: цей метод складається в згортанні критеріїв в один, що дозволяє розв'язати задачу одновимірної оптимізації. Зазвичай цей метод використовують для випадків, коли критерії пов'язані між собою та можуть бути згорнуті в один.

Враховуючи особливості кожного методу було обрано метод аналізу ієрархій (МАІ) для реалізації підсистеми визначення оптимального розподілу інцидентів при спрацюванні об'єкта. Метод аналізу ієрархій є ефективним методом вирішення задач багатокритеріальної оптимізації з кількома критеріями, оскільки дозволяє визначити вагомість кожного критерію та його вплив на кінцевий результат. Цей метод дає можливість структурувати проблему та порівнювати різні альтернативи, забезпечуючи зручний та логічний підхід до прийняття рішень. За допомогою МАІ можна визначити ієрархію критеріїв та їх підкритеріїв, а також вагомість кожного з них. Далі, використовуючи матриці порівнянь, можна визначити попарну важливість кожного критерію та підкритерію. Після цього можна розрахувати загальну вагомість кожного альтернативного рішення, що дозволяє визначити найкращий варіант [28].

Однією з переваг МАІ є можливість врахування як об'єктивних, так і суб'єктивних факторів, оскільки вагомість кожного критерію визначається експертами або користувачами. Крім того, МАІ може бути застосований до будь-якого типу проблеми багатокритеріальної оптимізації, включаючи проблеми зі змішаними типами критеріїв (наприклад, якісні та кількісні). Таким чином, МАІ є гнучким та ефективним методом вирішення задач багатокритеріальної оптимізації, який дозволяє враховувати різні критерії та визначати їх вагомість для прийняття оптимального рішення.

Процес вирішення проблеми складається з наступних кроків [29]:

- 1) Формування ієрархії критеріїв та їх підкритеріїв. Першим кроком є визначення всіх критеріїв, що необхідні для вирішення задачі багатокритеріальної оптимізації. Крім того, кожен критерій може мати свої підкритерії, які допомагають розширити множину критеріїв та врахувати більше факторів. Наприклад, якщо задача полягає у виборі автомобіля, то критерії можуть включати ціну, якість, комфортність, безпеку, екологічність та інші.

- 2) Порівняння критеріїв між собою за їх важливістю. Після того, як було сформовано ієрархію критеріїв, необхідно визначити їх відносну важливість

один відносно іншого. Цей процес зазвичай проводять експерти з відповідного фахового напрямку. Експерти оцінюють кожен критерій за шкалою від 1 до 9, де 1 означає, що два критерії мають однакову важливість, а 9 означає, що один критерій значно важливіший за інший. Шкалу порівняння наведено в таблиці 2.1.

Таблиця 2.1 – Шкала порівняння критеріїв

<b>Бал, k</b>	<b>Визначення</b>	<b>Пояснення</b>
1	Рівна важливість	Внесок до цілі однаковий
3	Помірно переважає	Досвід та судження трохи сприяють одному з критеріїв над іншим
5	Суттєво переважає	Досвід та судження сильно сприяють одному з критеріїв над іншим
7	Значна перевага	Перевага дуже сприятлива до одного з критеріїв, його домінування продемонстровано на практиці
9	Дуже велика перевага	Докази, що сприяють одному з критеріїв над іншим є найвищим можливим порядком підтвердження
2,4,6,8	Проміжні значення	Коли критерії дуже близькі, додавання знаків після коми дозволяє показати наявність різниці
1/k	Обернені значення	-

3) Побудова матриць парних порівнянь. На основі порівнянь вагомості критеріїв створюються матриці парних порівнянь. Кожна матриця відображає відносну важливість двох критеріїв або підкритеріїв. Якщо критеріїв  $n$ , то потрібно створити  $n$  матриць порівнянь. Матриці заповнюються значеннями від 1 до 9 відповідно до того, наскільки важливіший один критерій від іншого. Матриця повинна бути симетричною, тобто відносні значення вагомості між двома критеріями однакові, незалежно від того, який критерій є першим, а який другим.

4) Розрахунок вагових коефіцієнтів критеріїв. На основі матриць парних порівнянь розраховуються вагові коефіцієнти кожного критерію. Для цього знаходять власні вектори матриці, які відповідають її власним значенням, та нормалізують їх таким чином, щоб сума кожного вектора дорівнювала одиниці.

5) Розрахунок загальної оцінки кожного варіанту. Використовуючи отримані вагові коефіцієнти кожного критерію можна розрахувати загальну оцінку кожного варіанту, який оцінюється за допомогою ранжування та віднесення до відповідної категорії.

6) Вибір найкращого варіанту. На основі отриманих оцінок вибирається найкращий варіант, який задовольняє всі вимоги критеріїв та підкритеріїв.

## 2.4 Математична модель методу аналізу ієрархій

Метод аналізу ієрархій - це метод багатокритеріального прийняття рішень, який дозволяє порівнювати різні альтернативи на основі багатьох критеріїв і зважувати їх значимість відносно один одного. [30] Математична модель методу будується на основі матриць парних порівнянь, які визначають відносну важливість кожного критерію та альтернативи порівняно з іншими.

1) Створення задачі у вигляді ієрархічної структури (визначення цілі, критеріїв та альтернатив).

2) Попарне порівняння критеріїв. Нехай маємо набір критеріїв  $A_1 \dots A_n$ , тоді при порівнянні критеріїв  $A_i$  та  $A_j$  перевага оцінюється як  $a_{ij}$ , зворотне порівняння  $a_{ji} = \frac{1}{a_{ij}}$ , порівняння критерію із самим собою дорівнює одиниці. Таким чином маємо матрицю попарних порівнянь критеріїв:

$$\begin{array}{c} A_1 \dots A_n \\ \left[ \begin{array}{ccc} a_{1,1} & \dots & a_{1,n} \\ \dots & & \\ a_{n,1} & \dots & a_{n,n} \end{array} \right] \end{array} \quad (2.1)$$

3) Визначення вагових коефіцієнтів з попарних порівнянь критеріїв:

$$W_i = \frac{\sqrt[n]{\prod_{j=1}^n a_{i,j}}}{\sum_{i=1}^n \sqrt[n]{\prod_{j=1}^n a_{i,j}}} \quad (2.2)$$

4) Перевірка узгодженості матриці порівнянь критеріїв:

Побудова матриці  $E$ , розмірністю  $n \times n$ , елементи якої є:

$$\epsilon_{i,j} = a_{i,j} \frac{W_j}{W_i} \quad (2.3)$$

Обчислення усіх елементів матриці  $E$ , обираючи максимальне значення:

$$\lambda_{max} = \sum_{j=1}^n \epsilon_{i,j} \quad (2.4)$$

Обчислення індексу узгодженості:

$$\mu = \frac{|\lambda_{max} - n|}{n - 1} \quad (2.5)$$

Обчислення відношення узгодженості матриці порівнянь:

Якщо  $\frac{\mu}{\mu_T} < 0.1$ , де  $\mu_T$  – табличний індекс (табл. 2.2), то перейти до п'ятого пункту, інакше повернутись до 2 пункту і переглянути порівняння критеріїв.

Таблиця 2.2 – Значення табличного індексу відносно розміру матриці

Розмір матриці	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Табличний індекс	0	0	0,52	0,89	1,11	1,25	1,35	1,4	1,45	1,49	1,52	1,54	1,56	1,58	1,59

5) Розрахунок пріоритетів альтернатив. Спочатку необхідно провести попарне порівняння альтернатив за кожним критерієм, таким чином отримано  $K$  матриць попарного порівняння, з них отримуються  $K$  вектори ваг з  $n$  елементів кожен де  $(k)$  означає  $k$ -й критерій,  $n$  - кількість альтернатив:

$$W^{(k)} = \begin{bmatrix} W_1^{(k)} \\ \dots \\ W_n^{(k)} \end{bmatrix} \quad (2.6)$$

З векторів  $W^{(k)}$  отримується матриця, кожен рядок якої відповідає одній альтернативі:

$$\begin{matrix} A_1 \\ \dots \\ A_n \end{matrix} \begin{bmatrix} W_1^{(1)} & \dots & W_1^{(k)} \\ \dots & \dots & \dots \\ W_n^{(1)} & \dots & W_n^{(k)} \end{bmatrix} \quad (2.7)$$

Отриману матрицю необхідно помножити на вектор вагових коефіцієнтів критеріїв:

$$\begin{bmatrix} W_1^{(1)} & \dots & W_1^{(k)} \\ \dots & \dots & \dots \\ W_n^{(1)} & \dots & W_n^{(k)} \end{bmatrix} \begin{bmatrix} W^{(1)} \\ \dots \\ W^{(k)} \end{bmatrix} \quad (2.8)$$

Добутком є вектор пріоритетів альтернатив, таким чином, альтернатива з найбільшим значенням має найвищий пріоритет у виборі.



## 2.5 Методи формалізації алгоритму діяльності бригади

Для підсистеми визначення оптимальної бригади необхідно розрахувати ймовірності правильного й своєчасного виконання алгоритму діяльності всіх бригад, при чому для кожного рівня загрози заявки це значення буде відрізнятися через різні показники часу й правильності виконання окремих операцій. Для вирішення цієї задачі можна було б розглянути різні системи, що дають можливість описати процеси функціонування людино-машинних систем [31, 32, 33, 34]:

- мовно-алгебраїчні системи (PERT, GERT, МКШ мережі, мережі передування);
- логічні системи (формальні граматики, алгоритмічні моделі, мережі Петрі);
- алгебраїчні системи (марковські та напівмарковські процеси).

Але ці системи мають деякі недоліки, що унеможливають їх використання у вирішенні поставленої задачі, серед них: неможливість отримати кількісний показник при розрахунках ефективності виконання алгоритму діяльності, підтримка лише послідовно функціонуючих елементів, обмеження кількості ітерацій в циклах, складність розрахунків для отримання результату [35]. Таким чином було вирішено використовувати функціональні мережі як спосіб опису алгоритму діяльності.

Ідея методу функціональних мереж для опису людино-машинних систем полягає в тому що весь алгоритм розбивається на елементарні операції, кожна з яких має свої показники часу виконання та безпомилковості, такі операції в функціональній мережі називаються типовими одиницями (наведені в додатку Б). Потім відбувається редукція функціональної мережі, головна мета якої скоротити кількість операцій, зберігаючи при цьому функціональні характеристики системи. Процес заключається у пошуку певних комбінацій

типових структур (наведені в додатку В), які можна замінити на еквівалентну типову одиницю з одночасними розрахунками оновлених показників часу й безпомилковості виконання. Редукція вважається закінченою, якщо в описі системи залишилась одна типова одиниця, показники якої і є кінцевими показниками всієї системи [36, 37]. Приклад функціональної мережі та її редукцію зображено на рисунку 2.1. Реалізацію оцінювання ймовірності правильного й своєчасного виконання алгоритму діяльності бригад проводимо з використанням моделей наукової школи професора Анатолія Губінського [38, 39, 40] та програмного забезпечення, розробленого автором в бакалаврській роботі [6].

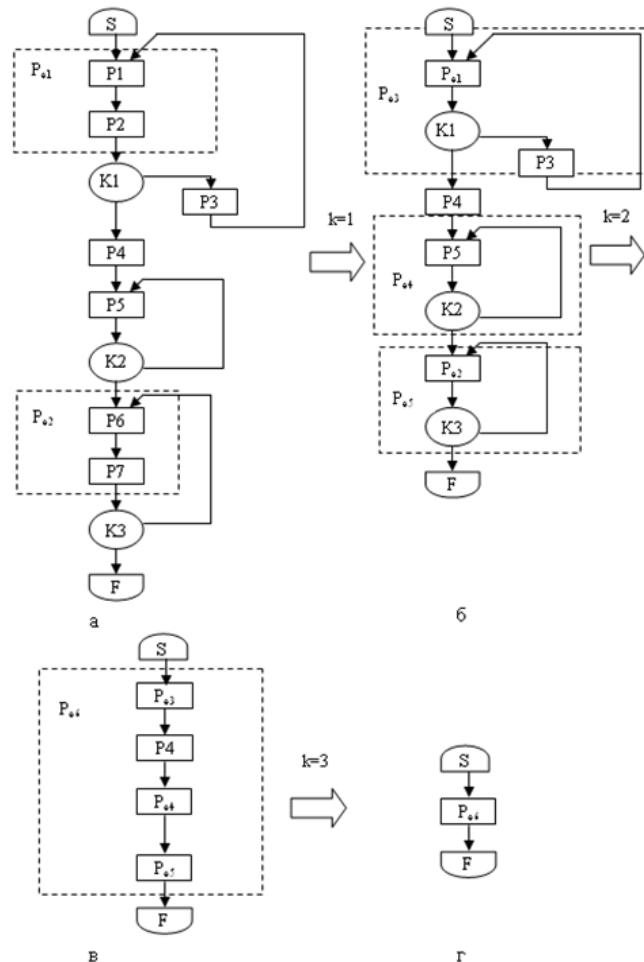


Рисунок 2.1 – Приклад редукції функціональної мереж

## 2.6 Вибір засобів реалізації

Для реалізації клієнтської частини інформаційної технології було обрано бібліотеку JavaScript – React. Вона побудована на основі компонентів - це означає, що інтерфейс додатку розбивається на невеликі, самодостатні частини, кожна з яких відповідає за свою функціональність. Компоненти можуть бути перевикористані в різних частинах додатку, що забезпечує ефективне використання коду і спрощує процес розробки. Ще одна важлива особливість React - це віртуальний DOM (Document Object Model), який дозволяє оновлювати тільки ті елементи на сторінці, які змінилися, не перезавантажуючи всю сторінку. Це забезпечує швидкість роботи додатку та зменшує навантаження на сервер. Ще однією зручною особливістю React є те, що можна працювати з багатьма іншими бібліотеками та фреймворками, це забезпечує більш широкий спектр можливостей для розробки веб-додатків.

Для реалізації компонентів що використовують мапи було обрано бібліотеку React Leaflet. Leaflet - це легкий веб-фреймворк для створення інтерактивних карт у веб-додатках. Він дозволяє додавати до карт різноманітні маркери, лінії, полігони, та інші елементи, а також використовувати різні тайлові служби для отримання даних про карти. React Leaflet надає можливість використовувати Leaflet разом з React, що дозволяє більш ефективно управляти картами та робити їх інтерактивними.

В якості тайлової служби було обрано використати послуги компанії Visicom. Вона спеціалізується на розробці та наданні геодезичних технологій та послуг, пропонує різноманітні рішення для роботи з геоданими, включаючи веб-карти, навігацію, маршрутизацію, аналіз даних та інше. Компанія дає можливість розробникам інтегрувати геодезичні технології у свої власні веб-додатки та сервіси використовуючи API (Application Programming Interface). Visicom API - це програмний інтерфейс, який надає доступ до функціональності геодезичних

технологій компанії і містить різні методи, що дозволяють взаємодіяти з даними про місцезнаходження, навігацією, маршрутизацією, визначати маршрут між точками, відображати геодані на картах, здійснювати пошук місць, а також виконувати аналітичні завдання з даними про місцезнаходження. Він також містить документацію та різноманітні інструменти для розробників, що дозволяють легко інтегрувати його в веб-додатки.

Для роботи з методом аналізу ієрархій обрано бібліотеку Analytic Hierarchy Process (АНП), вона дає можливість імпортувати вхідні дані для вирішення багатокритеріальної задачі й відповідає за обчислення результатів. Необхідно лише інтегрувати її в веб-додаток для можливості задавати довільні критерії, альтернативи та їх кількість.

Оскільки необхідно зберігати дуже багато даних, які пов'язані між собою, було обрано використати реляційну базу даних MySQL. Вона дозволяє зберігати дані в окремих таблицях та використовувати ключі для зв'язку між ними. Коли дані пов'язані між собою, вони можуть бути розділені на кілька таблиць, які пов'язані між собою за допомогою зовнішніх ключів. Це дозволяє ефективно зберігати та організовувати дані та забезпечувати їх цілісність.

Для реалізації серверної частини було використано гнучкий та легкий у використанні фреймворк Express. З його використанням реалізовані всі запити для отримання, редагування, видалення і додавання даних до бази даних.

Серед програмного забезпечення, яке необхідне для реалізації інформаційної технології, варто відзначити: Microsoft Excel – для реалізації підсистеми аналізу і прогнозу безпомилковості реалізації алгоритму діяльності бригад; MATLAB – для побудови дерева рішень для визначення рівня загрози заявок; MySQL Workbench – для роботи з базою даних; Postman – для тестування запитів; Git – система контролю версій; Visual Studio Code – редактор коду.

## 3 МОДЕЛЮВАННЯ Й ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

### 3.1 Структурно-функціональне проектування

IDEF0 (Integrated Definition for Function Modeling) - це методологія моделювання функцій та процесів, що використовується для розробки систем, що мають складну структуру та функціонування. Цей метод дозволяє аналізувати процеси та розбивати їх на більш малі компоненти, щоб краще зрозуміти взаємозв'язки між ними. IDEF0 може бути використана для визначення потреб в інформації, документації та інших ресурсів для забезпечення ефективного функціонування системи. Крім того, вона дозволяє розглядати процес з різних точок зору, що допомагає покращити його роботу та забезпечити оптимальний результат. У цілому, використання IDEF0 дозволяє забезпечити більш ефективну та оптимальну роботу системи, підвищити якість продукту та зменшити час та затрати на розробку [41]. Серед елементів що повинні бути в IDEF0 діаграмі виділяють наступні:

- Об'єкти, які беруть участь у процесі, зображуються у вигляді прямокутників;
- Входи та виходи - дані, які входять або виходять з процесу. Вони зображуються як стрілки, які вказують напрямок потоку даних;
- Механізми - обладнання, програмне забезпечення або будь-які інші засоби, які використовуються для виконання функцій;
- Керування - керуючі елементи, які відповідають за порядок виконання функцій. Керування зображуються у вигляді стрілок, які вказують напрямок контролю або координації процесу.

IDEF0 діаграма проекту зображена на рисунку 3.1. Об'єктом є інформаційна технологія підтримки діяльності оператора пульта спостереження

за складними об'єктами. Входами в об'єкт є база даних з усією інформацією про об'єкти, що охороняються, бригади, патрульних, власників, інцидентів і т.д., значення показників інцидентів, елементарні операції діяльності бригад та показники ефективності виконання операцій бригадами. Механізмами є програмне й апаратне забезпечення та оператор. Керується процес за допомогою дерева рішень, методу аналізу ієрархій, сервісом Visicom, що надає API для роботи з мапами і маршрутами, та апаратом функціональних мереж Губінського.

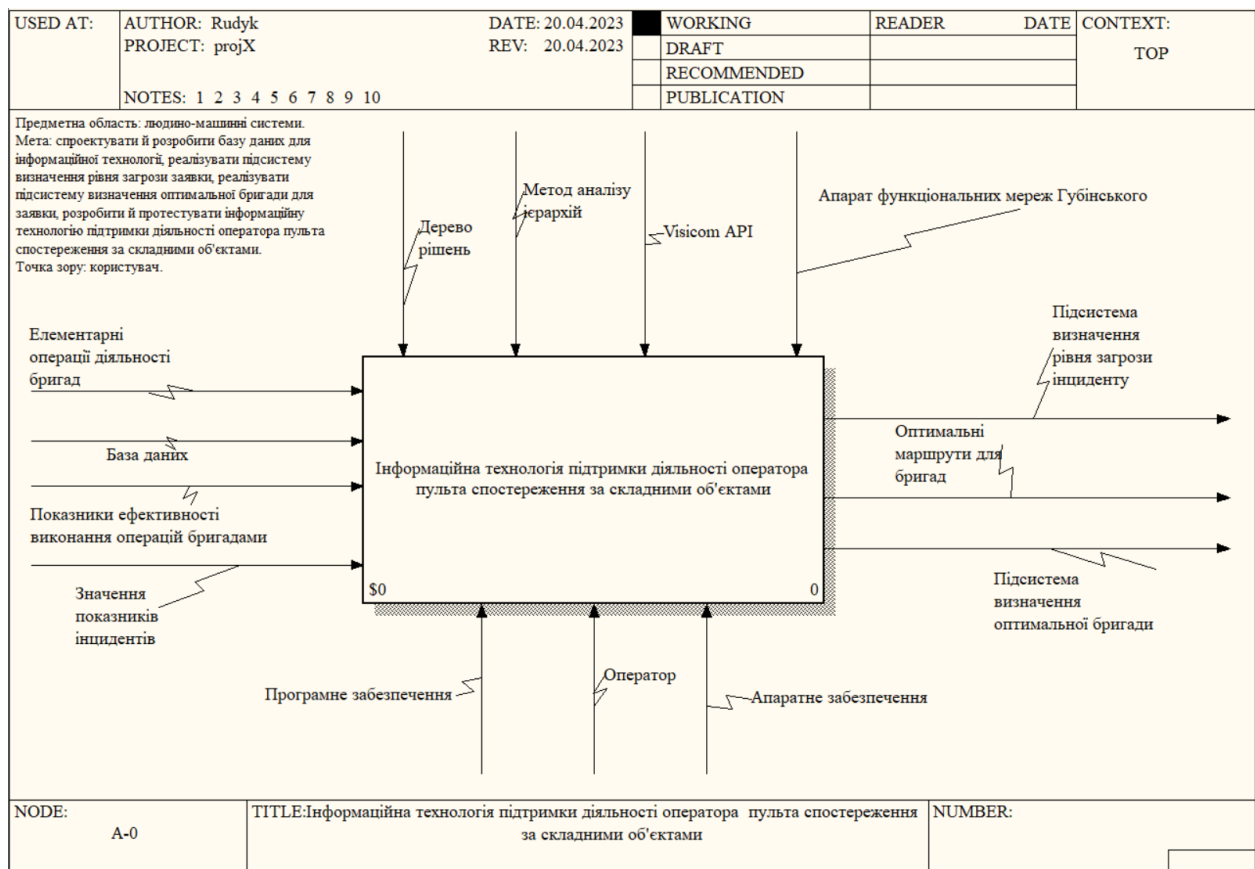


Рисунок 3.1 - IDEF0 діаграма проекту

В декомпозиції нижнього рівня діаграми (рис 3.2) було розбито основний процес проекту на чотири компоненти такі як:

– Отримання сигналу про виникнення інциденту, на виході є безпосередньо сам інцидент з інформацією про місце, час та іншими характеристиками.

- Визначення рівня загрози інциденту, на виході отримується інцидент вже з визначеним рівнем загрози, який може бути наднизький, низький, середній, високий.
- Побудова оптимальних маршрутів для доступних бригад, на виході є зображений маршрут від місця розташування бригади до об'єкту де виник інцидент.
- Визначення оптимальної бригади для вирішення інциденту. Таким чином на виході отримуються підсистема визначення рівня загрози інциденту, оптимальні маршрути для бригад та підсистема визначення оптимальної бригади.

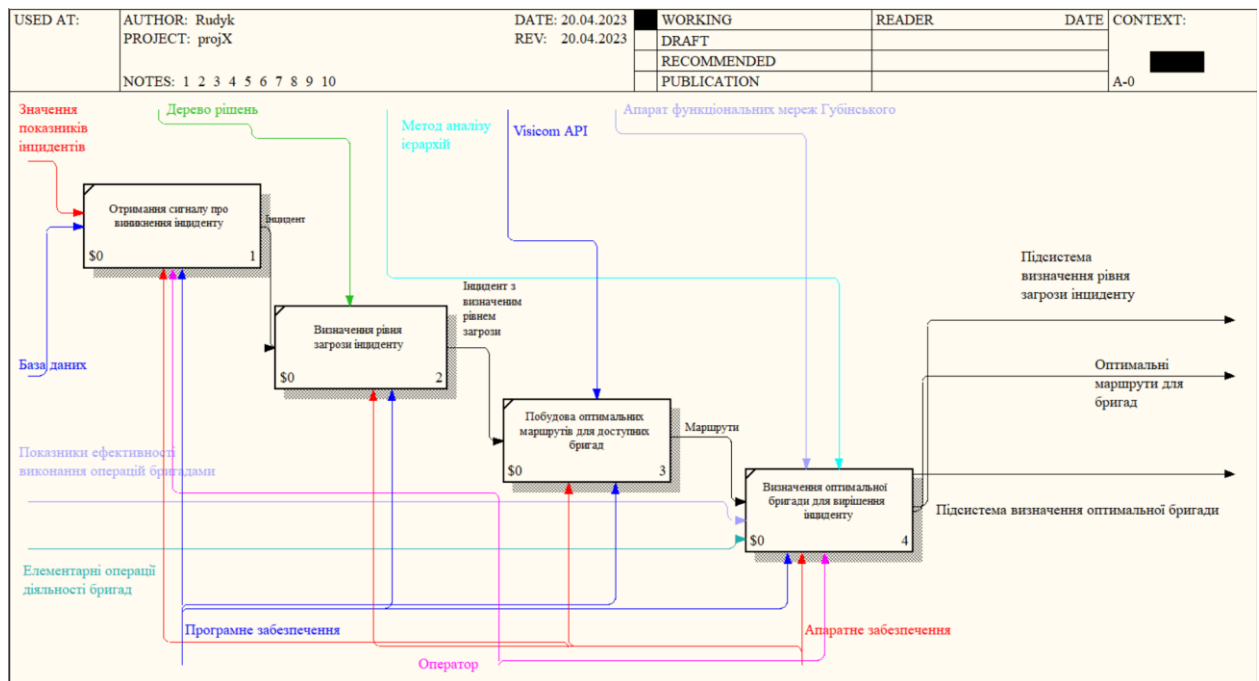


Рисунок 3.2 – Декомпозиція діаграми IDEF0

### 3.2 Моделювання варіантів використання

UML (Unified Modeling Language) - це мова моделювання, яка дозволяє створювати графічні моделі систем та програмного забезпечення. Однією з

найбільш поширених форм UML є діаграми, які надають графічне представлення структури та поведінки системи або програми. UML діаграми можуть бути використані для різних цілей [42], таких як:

- Моделювання структури системи: діаграми класів UML використовуються для моделювання класів, їх взаємозв'язків та атрибутів. Ці діаграми допомагають розуміти, як компоненти системи пов'язані між собою, а також їх характеристики.

- Моделювання поведінки системи: діаграми послідовності UML дозволяють моделювати взаємодію об'єктів системи та порядок виконання дій між ними. Ці діаграми допомагають розуміти, як система поводить себе в різних сценаріях взаємодії.

- Моделювання взаємодії між компонентами системи: діаграми компонентів UML використовуються для моделювання взаємодії між компонентами системи та їх взаємозв'язків. Ці діаграми допомагають розуміти, як компоненти взаємодіють між собою та які сервіси вони надають.

- Моделювання розгортання системи: діаграми розгортання UML дозволяють моделювати архітектуру системи на рівні фізичних компонентів, таких як сервери, бази даних та інші. Ці діаграми допомагають розуміти, як система розгорнута та які компоненти взаємодіють між собою.

UML діаграми також є стандартним засобом комунікації між різними стейкхолдерами, такими як розробники, тестувальники та бізнес-аналітики. Вони можуть допомогти уникнути непорозумінь та забезпечити, що всі стейкхолдери розуміють те ж саме щодо структури та поведінки системи.

Нарешті, UML діаграми є ефективним засобом документування проекту програмного забезпечення. Вони можуть бути використані для зберігання інформації про структуру та поведінку системи, яка може бути використана в майбутньому для підтримки та розширення системи.

Отже, UML діаграми є важливим засобом моделювання та візуалізації структури та поведінки системи або програми. Вони допомагають розробникам



та стейкхолдерам зрозуміти, як система пов'язана та як вона поводить себе, що дозволяє ефективніше розробляти, тестувати та підтримувати програмне забезпечення. UML діаграма проекту зображена на рисунку 3.3

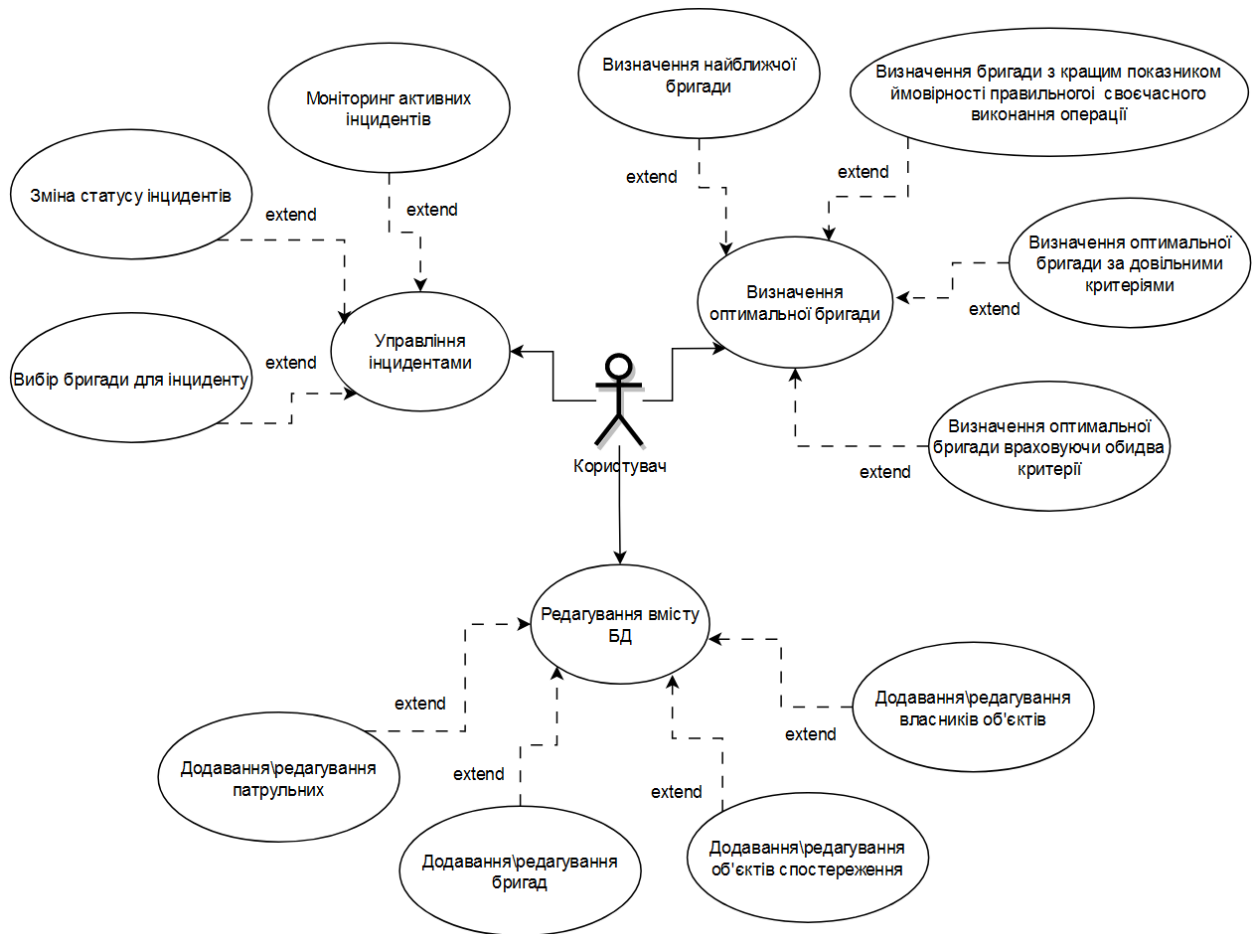


Рисунок 3.3 – UML діаграма проекту

### 3.3 Проектування моделі бази даних

Для виконання задач інформаційної технології підтримки діяльності оператора пульта спостереження за складними об'єктами база даних повинна зберігати інформацію про об'єкти спостереження та їх власників, патрульних та бригад, що складаються з двох різних патрульних. Також необхідно зберігати інформацію про всі інциденти що виникають з історією їх опрацювань, статусами, часом, бригадою, що займалась ними і т.д. Враховуючи описане вище,

створено логічну модель бази даних (рис 3.4) з усіма необхідними таблицями та полями для забезпечення функціонування інформаційної технології.



Рисунок 3.4 – Логічна модель бази даних

## 4 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПІДТРИМКИ ДІЯЛЬНОСТІ ОПЕРАТОРІВ ЦЕНТРАЛІЗОВАНИХ ПУЛЬТІВ СПОСТЕРЕЖЕННЯ ЗА ОБ'ЄКТАМИ

### 4.1 Реалізація користувальницького інтерфейсу

Перед реалізацією всіх підсистем інформаційної технології було створено середовище роботи оператора з усім необхідним функціоналом, а саме: загальну сторінку з мапою міста, бригадами та об'єктами (рис 4.1), що знаходяться під охороною; сторінки для додавання або редагування об'єктів (рис 4.2), клієнтів (рис. 4.3), патрульних (рис 4.4) і бригад (рис 4.5).

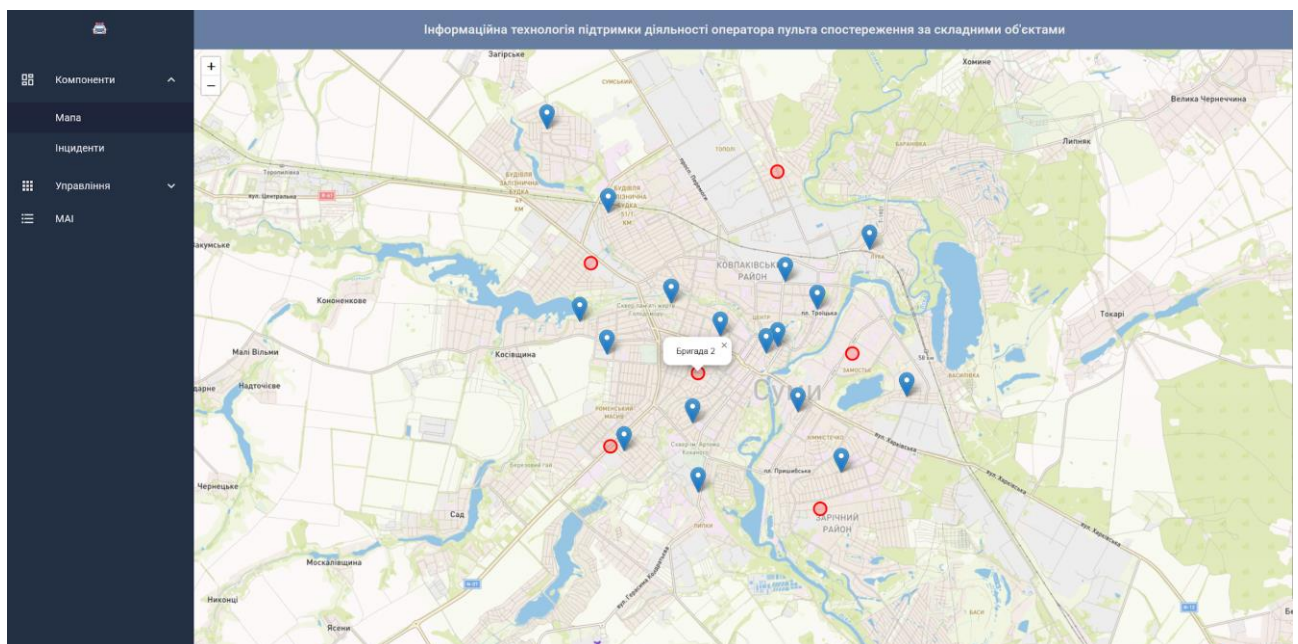


Рисунок 4.1 - Загальна сторінка з мапою міста

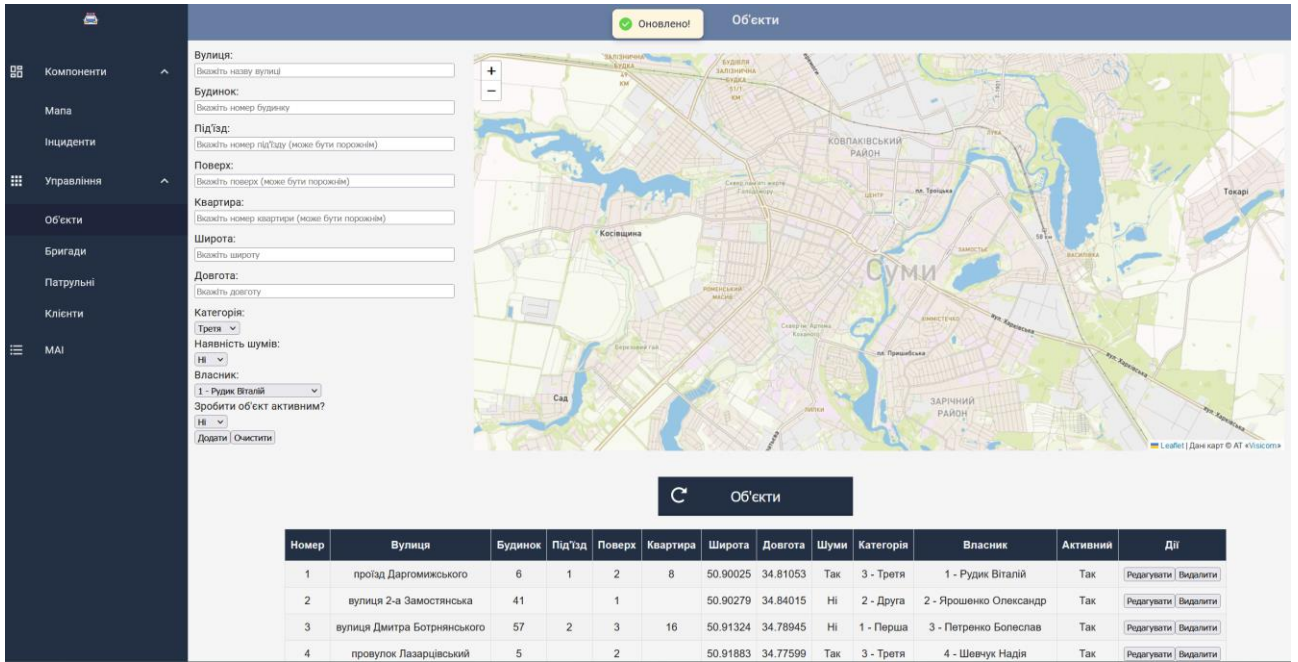


Рисунок 4.2 – Сторінка “Об’єкти”

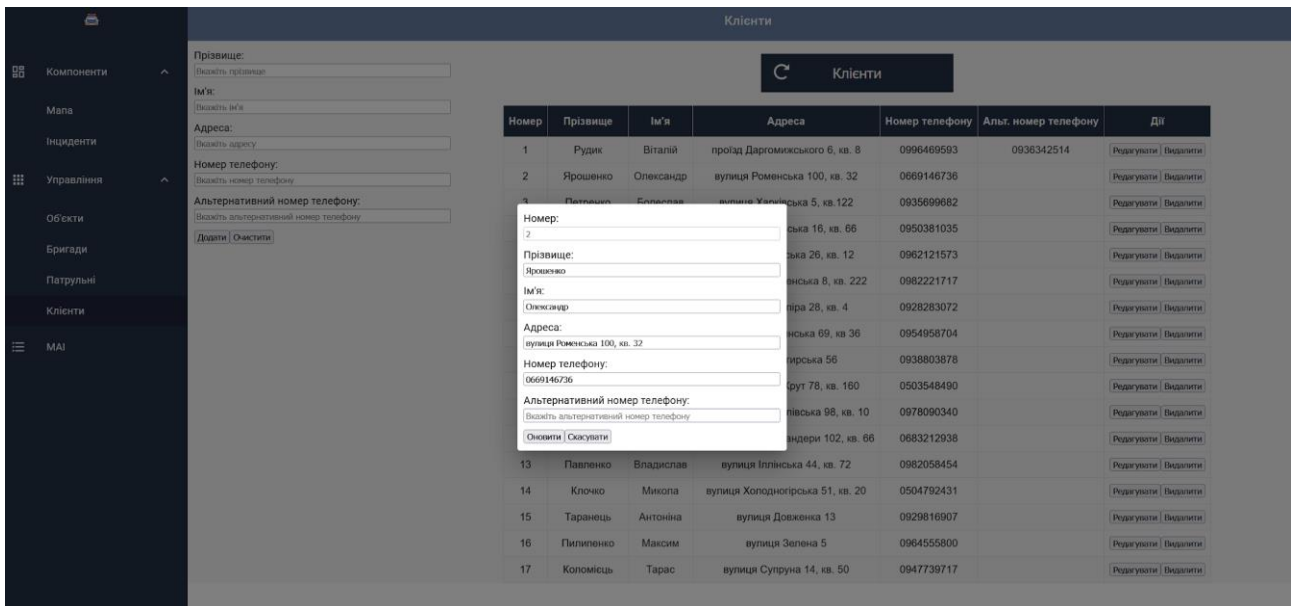


Рисунок 4.3 – Редагування даних на сторінці “Клієнти”

Патрульні

ПІБ:

Дата народження:

Дата початку роботи:

[Додати](#) [Очистити](#)

Патрульні

Номер	ПІБ	Дата народження	Стаж	Дії
1	Погорілий Дмитро Іванович	07/07/1999	05/05/2021	<a href="#">Редагувати</a> <a href="#">Видалити</a>
2	Ігнатенко Денис Степанович	04/01/1975	02/11/2004	<a href="#">Редагувати</a> <a href="#">Видалити</a>
3	Філатов Арсеній Андрійович	24/02/1990	22/12/2018	<a href="#">Редагувати</a> <a href="#">Видалити</a>
4	Бенедюк Артем Ігорович	16/03/1993	14/01/2016	<a href="#">Редагувати</a> <a href="#">Видалити</a>
5	Боль Антон Михайлович	23/11/1983	21/09/2013	<a href="#">Редагувати</a> <a href="#">Видалити</a>
6	Слесар Сергій Анатолійович	11/04/1987	09/02/2015	<a href="#">Редагувати</a> <a href="#">Видалити</a>
7	Яцик Олександр Васильович	29/05/1979	27/03/2007	<a href="#">Редагувати</a> <a href="#">Видалити</a>
8	Царенко Антон Миколайович	13/06/1985	11/04/2017	<a href="#">Редагувати</a> <a href="#">Видалити</a>
9	Горбач Ігор Віталійович	07/12/1988	05/10/2011	<a href="#">Редагувати</a> <a href="#">Видалити</a>
10	Федоров Сергій Русланович	19/08/1981	17/06/2019	<a href="#">Редагувати</a> <a href="#">Видалити</a>
11	Венглинський Олег Валентинович	28/03/1996	26/01/2020	<a href="#">Редагувати</a> <a href="#">Видалити</a>
12	Мороз Олександр Макарович	04/09/1991	02/07/2016	<a href="#">Редагувати</a> <a href="#">Видалити</a>
13	Романов Данило Юрійович	22/07/1984	20/05/2012	<a href="#">Редагувати</a> <a href="#">Видалити</a>
14	Марчук Петро Ярославович	18/01/1981	16/11/2012	<a href="#">Редагувати</a> <a href="#">Видалити</a>
15	Кос Геннадій Владиславович	27/05/1978	25/03/2007	<a href="#">Редагувати</a> <a href="#">Видалити</a>
16	Шевченко Віталій Володимирович	15/06/1986	13/04/2013	<a href="#">Редагувати</a> <a href="#">Видалити</a>

Рисунок 4.4 – Сторінка “Патрульні”

Бригади

Перший патрульний:

Другий патрульний:

Широта:

Довгота:

Ймовірність правильного і своєчасного виконання операції для наднизького рівня загрози:

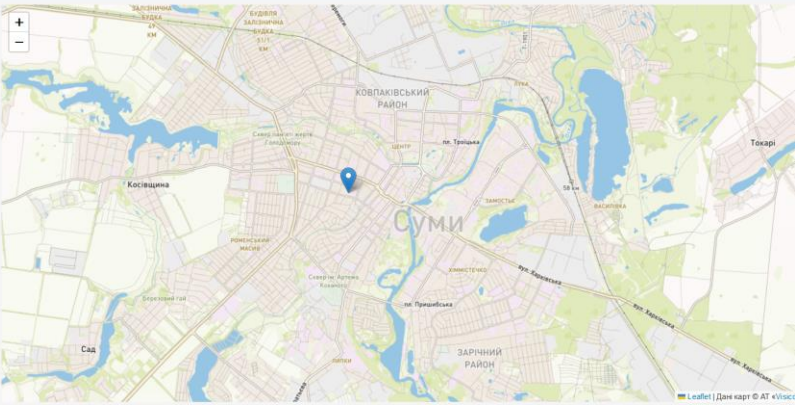
Ймовірність правильного і своєчасного виконання операції для низького рівня загрози:

Ймовірність правильного і своєчасного виконання операції для середнього рівня загрози:

Ймовірність правильного і своєчасного виконання операції для високого рівня загрози:

Зробити бригаду активною?

[Додати](#) [Очистити](#)



Бригади

Номер	Перший патрульний	Другий патрульний	Широта	Довгота	Ефективність нн	Ефективність н	Ефективність с	Ефективність в	Активна	Дії
1	1 - Погорілий Дмитро Іванович	2 - Ігнатенко Денис Степанович	50.91027	34.82542	68	66	65	63	Так	<a href="#">Редагувати</a> <a href="#">Видалити</a>
2	3 - Філатов Арсеній Андрійович	4 - Бенедюк Артем Ігорович	50.90695	34.78348	67	67	66	64	Так	<a href="#">Редагувати</a> <a href="#">Видалити</a>

Рисунок 4.5 – Сторінка “Бригади”

## 4.2 Реалізація підсистеми визначення рівня загрози

Однією з проблем діяльності оператора є визначити ступінь загрози інциденту, що надійшов для його подальшого опрацювання. Від рівня загрози залежить технологія усунення інциденту і дані для моделювання діяльності оперативних бригад. Поставлена задача створити класифікатор на основі технологій інтелектуального аналізу даних. В якості вхідних даних є статистичні дані щодо показників інцидентів та рівень загрози що їм відповідає. Ці дані були використані як навчальна матриця (табл. 4.1) для створення класифікатора рівня загрози інциденту.

Таблиця 4.1 – Навчальна матриця

Категорія об'єкту	Кількість зон спрацювання	Кількість повторних спрацювань	Наявність об'єктів, що створюють шуми	Рівень загрози
1	5	8	0	Високий
2	1	4	0	Середній
3	2	2	1	Низький
3	1	0	1	Наднизький
2	4	6	1	Високий
1	1	0	0	Низький
3	1	6	1	Середній
2	1	1	1	Низький
3	5	5	0	Високий
2	1	2	0	Середній
3	3	0	1	Низький
1	3	0	0	Високий
2	1	0	1	Наднизький
2	2	1	1	Низький
1	1	0	0	Наднизький
2	2	0	1	Наднизький
2	3	0	0	Середній
2	3	1	1	Низький
3	2	1	1	Низький
1	2	1	0	Високий
3	2	1	0	Середній
2	1	1	0	Середній
3	1	1	0	Низький

Далі в Matlab було введено (рис 4.6) навчальну матрицю й отримано дерево рішень (рис 4.7), яке використовується при визначенні рівня загрози інцидентів.

```

P = [1 5 8 0; 2 1 4 0; 3 2 2 1; 3 1 0 1; 2 4 6 1; 1 1 0 0; 3 1 6 1; 2 1 1 1; 3 5 5 0; 2
1 2 0; 3 3 0 1; 1 3 0 0; 2 1 0 1; 2 2 1 1; 1 1 0 0; 2 2 0 1; 2 3 0 0; 2 3 1 1; 3 2 1 1;
1 2 1 0; 3 2 1 0; 2 1 1 0; 3 1 1 0]
R = ["Високий" "Середній" "Низький" "Наднизький" "Високий" "Низький" "Середній"
"Низький" "Високий" "Середній" "Низький" "Високий" "Наднизький" "Низький" "Наднизький"
"Наднизький" "Середній" "Низький" "Низький" "Високий" "Середній" "Середній" "Низький"]
t = classregtree(P, R, 'names', {'Категорія' 'Зони' 'Повтори' 'Шуми'})
view(t)
dtoclass = t.eval(P);
bad = ~strcmp(dtoclass,R);
Err = sum(bad)/(size(R,1))

```

Рисунок 4.6 – Реалізація в Matlab

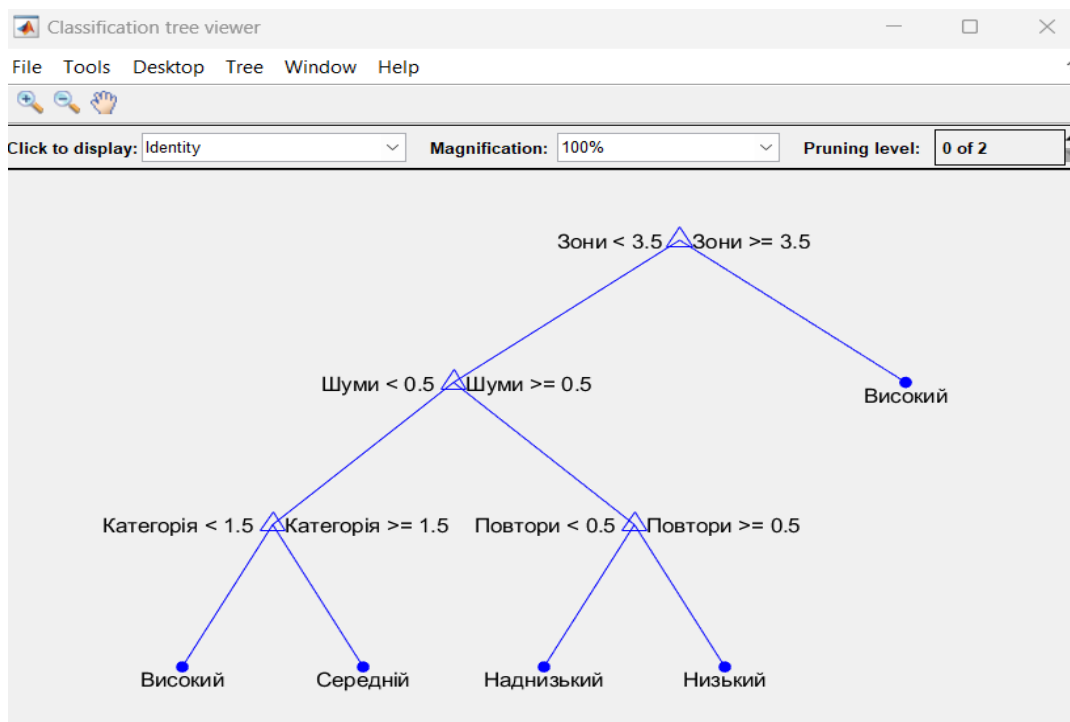


Рисунок 4.7 – Дерево рішень

Отриманий класифікатор було реалізовано з використанням конструкції `if...else` на сторінці інцидентів (рис 4.8). Коли виникає новий інцидент, він має лише такі показники як: кількість зон (кімнат або датчиків, що спрацювали), категорія (в залежності від важливості об'єкту), кількість повторних спрацювань (кількість спрацювань після початкового), наявність об'єктів, що створюють шуми (тварини, або інші чинники, що можуть призвести до фейкових

спрацювань). Далі завдяки створеному класифікатору ці дані аналізуються та визначається рівень загрози з подальшим записом до бази даних (рис 4.9).

Інциденти

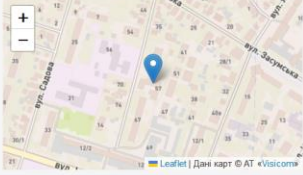
**В процесі**

Інцидент | Номер 3 | 01.02.2023 15:31:02 | Зони 3 | Повторень 1 | Категорія 1 | Координати (50.91, 34.79)

Об'єкт | Номер 3 | вулиця Дмитра Ботнянського 57, під'їзд 2, поверх 3, квартира 16

Клієнт | Номер 3 | Петренко Болеслав | вулиця Харківська 5, кв.122 | 0935699682

Рівень загрози - Високий



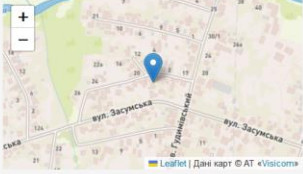
**В процесі**

Інцидент | Номер 4 | 01.02.2023 15:31:02 | Зони 1 | Повторень 1 | Категорія 3 | Координати (50.92, 34.78)

Об'єкт | Номер 4 | провулок Лазарцівський 5, поверх 2 | 6 тварини

Клієнт | Номер 4 | Шевчук Надія | вулиця Покровська 16, кв. 66 | 0950381035

Рівень загрози - Високий



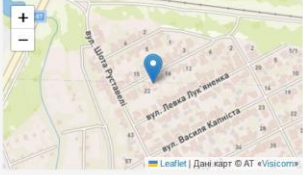
**В процесі**

Інцидент | Номер 5 | 01.02.2023 15:31:02 | Зони 3 | Повторень 3 | Категорія 2 | Координати (50.93, 34.76)

Об'єкт | Номер 5 | вулиця Волошкова 20, поверх 1

Клієнт | Номер 5 | Мірошніченко Василь | вулиця Засумська 26, кв. 12 | 0962121573

Рівень загрози - Середній



**В процесі**

Інцидент | Номер 7 | 01.02.2023 15:31:02 | Зони 3 | Повторень 3 | Категорія 2 | Координати (50.92, 34.81)

Об'єкт | Номер 7 | вулиця Пришвина 2, поверх 1

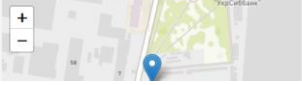


Рисунок 4.8 – Сторінка “Інциденти”

Limit to 1000 rows

```
1 • SELECT * FROM operator_helper.incidents;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap C

	id	object_id	timestamp_id	is_dosed	is_checked	iterations	zones	danger
▶	1	1	1	0	1	1	1	Низький
	2	2	1	0	1	3	2	Середній
	3	3	1	0	1	1	3	Високий
	4	4	1	0	1	1	1	Високий
	5	5	1	0	1	3	3	Середній
	6	6	1	0	1	5	5	Високий
	7	7	1	0	1	3	3	Середній
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 4.9 – Таблиця з інцидентами в базі даних



### 4.3 Реалізація підсистеми оцінки часу транспортування бригад на місце події

Оскільки час є дуже важливим показником при виконанні дій, спрямованих на усунення інцидентів, було реалізовано підсистему оцінки часу транспортування бригад на місце події. Завдяки цій підсистемі оператору не треба думати над тим, якій бригаді назначити усунення інциденту що виник. Для цього використовується Visicom API, а саме запит distance (рис 4.10), який в якості вхідних даних приймає координати двох точок, в нашому випадку це координати місця знаходження бригади та координати об'єкту, де виникло спрацювання, на виході (рис 4.11) отримується GeoJSON-об'єкт, що містить такі поля як properties (з вхідними координатами двох точок і дистанцією в метрах між ними) та geometry (об'єкти типу LineString, завдяки яких будується маршрут на мапі).

Таким чином, ефективність роботи оператора збільшується за рахунок автоматичного аналізу можливості виїзду і часу транспортування бригад на місце події, оскільки можна обрати оптимальну бригаду за критерієм часу прибуття до об'єкту відразу після виникнення інциденту. На рисунку 4.12 зображено приклад роботи реалізованої підсистеми.

```
const fetchRoutes = () => {
  try {
    const API_KEY = process.env.REACT_APP_VISICOM_API_KEY;
    const destination = [incident.objectlongitude, incident.objectlatitude]

    patrols.map(async patrol => {
      const origin = [patrol.longitude, patrol.latitude]
      var probability;
      const response = await axios.get("https://api.visicom.ua/data-api/5.0/core/distance.json?origin=" + origin + "&destination=" + destination)
      if (incident.incidentdanger === "Наднизький") {
        probability = patrol.patrolprobabilitysuperlow;
      }
      if (incident.incidentdanger === "Низький") {
        probability = patrol.patrolprobabilitylow;
      }
      if (incident.incidentdanger === "Середній") {
        probability = patrol.patrolprobabilitymedium;
      }
      if (incident.incidentdanger === "Високий") {
        probability = patrol.patrolprobabilityhigh;
      }
      const newRoute = { patrolid: patrol.id, response: response.data, distance: response.data.properties.distance, probability: probability, da
      setRoutes(r => [...r, newRoute])
      console.log(newRoute)
    })
  }
}
```

Рисунок 4.10 – Фрагмент реалізації запиту distance

GET <https://api.visicom.ua/data-api/5.0/core/distance.json?origin=34.783481,50.906953&destination=34.775991,50.918974&geometry=path&mode=driving-shortest&key=>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> origin	34.783481,50.906953	
<input checked="" type="checkbox"/> destination	34.775991,50.918974	
<input checked="" type="checkbox"/> geometry	path	
<input checked="" type="checkbox"/> mode	driving-shortest	

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "type": "Feature",
3    "properties": {
4      "distance": 1984,
5      "origin": {
6        "type": "Point",
7        "coordinates": [
8          34.783481,
9          50.906953
10       ]
11     },
12     "destination": {
13       "type": "Point",
14       "coordinates": [
15         34.775991,
16         50.918974
17       ]
18     }
19   },
20   "geometry": {
21     "type": "LineString",
22     "coordinates": [
23       [
24         34.783481,
25         50.906953
26       ],
27       [
28         34.783205,
29         50.906977

```

Рисунок 4.11 – Відповідь запиту distance

Поточний статус: відсутній  
 Бригада: відсутня  
 Змінити статус: 1 - Взято до опрацювання й направлено бригаду  
 Обрати бригаду: 2 - Філатов Арсеній Андрійович, Бенедок Артем Ігорович  
 Додати

Обрати найближчу бригаду  
 Обрати бригаду з кращим показником ймовірності своєчасного та правильного виконання  
 Обрати оптимальну бригаду враховуючи обидва критерії  
 Обрати оптимальну бригаду за довільними критеріями

Бригада: 2, Час: 90с, Відстань: 1984м, Ймовірність: 64%  
 Бригада: 6, Час: 102с, Відстань: 2241м, Ймовірність: 65%  
 Бригада: 5, Час: 154с, Відстань: 3391м, Ймовірність: 71%  
 Бригада: 3, Час: 196с, Відстань: 4310м, Ймовірність: 68%  
 Бригада: 1, Час: 252с, Відстань: 5546м, Ймовірність: 63%  
 Бригада: 4, Час: 286с, Відстань: 6286м, Ймовірність: 68%

Рисунок 4.12 – Приклад роботи підсистеми оцінки часу транспортування бригад на місце події

## 4.4 Реалізація підсистеми аналізу і прогнозу безпомилковості реалізації алгоритму діяльності бригад

Для реалізації підсистеми аналізу і прогнозу безпомилковості реалізації алгоритму діяльності бригад була використана розроблена мною раніше інформаційні технологія, яка дає можливість розраховувати показники часу та ймовірності виконання алгоритмів діяльності описаних у вигляді функціональної мережі. Для зручної і швидкої оцінки алгоритмів функціонування, підсистема має довідники типових функціональних структур, типових функціональних одиниць та довідник операцій. Таким чином, при введенні функціональної мережі необхідно обирати готові операції без введення їх показників часу і ймовірності. На рисунку 4.13 зображено загальну структуру підсистеми.

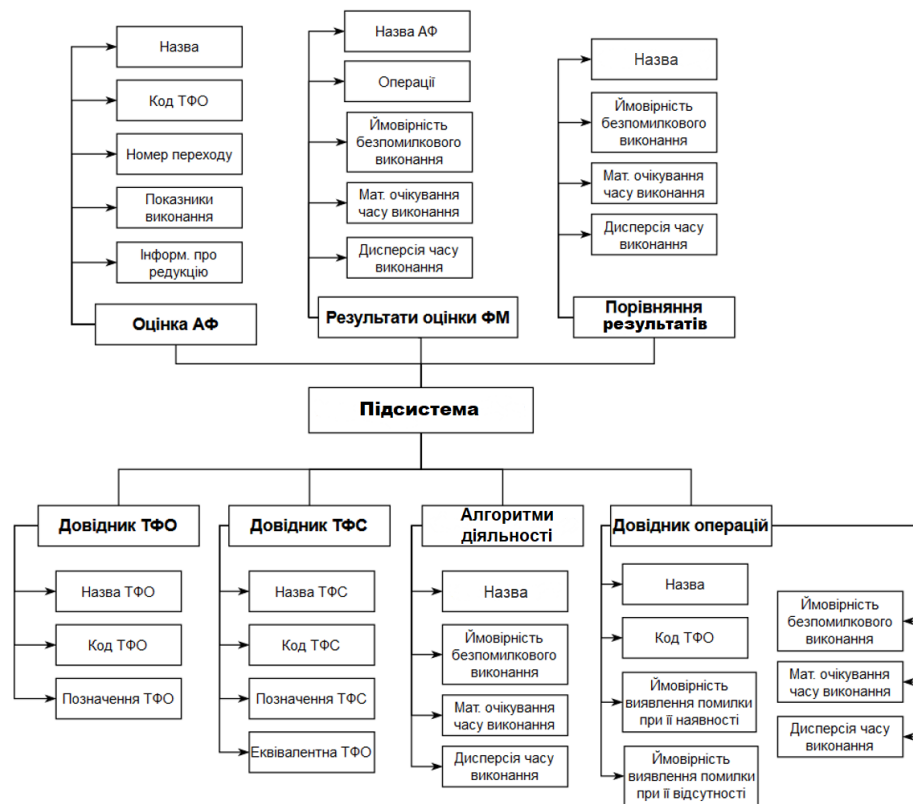


Рисунок 4.13 – Структура підсистеми аналізу і прогнозу безпомилковості реалізації алгоритму діяльності бригад

Оскільки основною задачею підсистеми є оцінка функціональної мережі, то необхідно було визначити основні операції які виконує бригада, побудувати функціональну мережу, ввести її в підсистему, й розрахувати показники ефективності виконання бригадами для кожного з можливих рівнів загрози (наднизький, низький, середній та високий). Таким чином, для розрахунку часових та ймовірнісних показників виконання алгоритму діяльності бригад, було проаналізовано їх роботу та визначено наступні елементарні операції:

- Отримати інформацію про інцидент по рації (P1);
- Повідомити оператору про успішне отримання інформації про інцидент (P2);
- Приїхати до місця інциденту (P3);
- Повідомити оператору про прибуття (P4);
- Оглянути об'єкт ззовні (P5);
- Повідомити оператору про результати огляду об'єкту (P6);
- Оглянути об'єкт всередині (P7);
- Повідомити оператору про результати огляду об'єкту (P8);
- Виявити причину інциденту (P9);
- Контроль правильності виявлення причини інциденту (K1);
- Проведення заходів оперативного реагування (P10);
- Повідомити оператору причину виникнення інциденту (P11);
- Перезавантажити пристрій спостереження (P12);
- Контроль над тим що пристрій перезавантажено (K2);
- Повідомити оператору при перезавантаженні пристрою спостереження (P13);
- Перезачинити об'єкт (P14);
- Контроль над тим що об'єкт зачинено (K3);
- Повідомити оператору про закриття інциденту (P15).

На рисунку 4.14 зображено функціональну мережу алгоритму діяльності бригад по усуненню інцидентів.

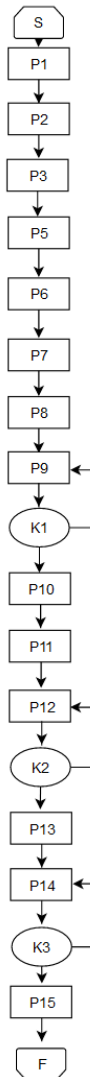


Рисунок 4.14 – ФМ алгоритму діяльності бригад по усуненню інцидентів

Цю ФМ необхідно було ввести в підсистему для проведення редукції та розрахунку часових та надійнісних показників всього алгоритму. Далі на скріншотах буде показано приклад оцінки алгоритму усунення інциденту однієї з бригад при наднизькому рівні загрози. Для цього в головному меню підсистеми (рис. 4.15) необхідно обрати “Введення опису ФМ”. Далі в формі що відкрилась треба по черзі ввести кожну операцію алгоритму з її показниками (рис 4.16), оскільки на листі “Довідник операцій” вже є інформація про кожну операцію

(рис 4.17), то внизу форми можна вибирати потрібну операцію і її показники автоматично з'являться у відповідних полях, це спрощує і пришвидшує процес введення опису ФМ для її подальшої оцінки.

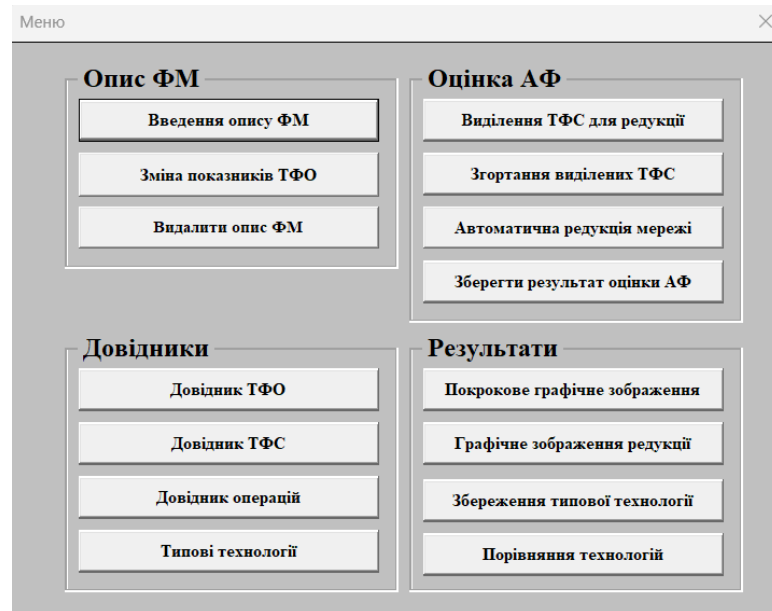


Рисунок 4.15 – Головне меню

The screenshot shows a window titled "Введення опису ФМ" (FM description) with a close button (X) in the top right corner. The form is divided into several sections:

- Ідентифікація технологій (Technology identification):**
  - Назва системи: ЦПС
  - Назва підсистеми: Бригада 5, наднизький рівень
  - Назва процесу: Усунення інциденту
  - Номер рядка таблиці для початку опису: 15
- Типова функціональна одиниця (Standard functional unit):**
  - Номер п/п: 10
  - Назва ТФО: 1 - Робоча
  - Код ТФО: 1
  - Позначення ТФО: р9
- Показники виконання ТФО (TFO execution indicators):**
  - Ймовірність виконання робочої операції без помилки (B1): .92
  - Ймовірність того що фактично правильне виконання буде визнане правильним(K11/P11): 0
  - Ймовірність того що фактично неправильне виконання буде визнане неправильним(K00/P00): 0
  - Математичне очікування часу виконання операцій(M(T)): 60
  - Дисперсія часу виконання(D(T)): 15
- Перехід (продовження функціонування) (Transition (continuation of operation)):**
  - При послідовному виконанні операцій алгоритму: 2
  - При невиконанні умов по КФ: 0
  - При невиконанні умов по КР: 0
  - На продовження цикла: 0
  - Номер на вихід з цикла: 0
  - Кількість повторів в циклі: 0
- Вибрати оператора і тип операцій (Select operator and operation type):** Бригада 5, наднизький рівень загрози
- Вибрати операцію (Select operation):** Виявити причину інциденту

At the bottom of the form, there are five buttons: "Попередня" (Previous), "Наступна" (Next), "Зберегти" (Save), "Додати" (Add), and "Вихід" (Exit).

Рисунок 4.16 – Форма введення ФМ

Бригада №5, наднизький рівень загрози						
Назва	ТФО	$B_1^{1'}$	$K_1^{11}$	$K_1^{00}$	$M(T)$	$D(T)$
Отримати інформацію про інцидент по рації	1 - Робоча	0,98	0	0	5	2
Повідомити оператору про успішне отримання інформації про інцидент	1 - Робоча	0,99	0	0	3	1
Приїхати до місця інциденту	1 - Робоча	0,95	0	0	180	60
Повідомити оператору про прибуття	1 - Робоча	0,97	0	0	5	2
Оглянути об'єкт ззовні	1 - Робоча	0,95	0	0	150	30
Повідомити оператору про результати огляду об'єкту	1 - Робоча	0,97	0	0	5	2
Оглянути об'єкт всередині	1 - Робоча	0,95	0	0	120	30
Повідомити оператору про результати огляду об'єкту	1 - Робоча	0,97	0	0	5	2
Виявити причину інциденту	1 - Робоча	0,92	0	0	60	15
Контроль правильності виявлення причини інциденту	3 - Контроль функціонування	0	0,965	0,965	10	3
Проведення заходів оперативного реагування	1 - Робоча	0,9	0	0	60	20
Повідомити оператору причину виникнення інциденту	1 - Робоча	0,97	0	0	5	2
Перезавантажити пристрій спостереження	1 - Робоча	0,96	0	0	30	10
Контроль над тим що пристрій перезавантажено	3 - Контроль функціонування	0	0,98	0,98	5	2
Повідомити оператору при перезавантаженні пристрою спостереження	1 - Робоча	0,97	0	0	5	2
Перезачинити об'єкт	1 - Робоча	0,99	0	0	40	15
Контроль над тим що об'єкт зачинено	3 - Контроль функціонування	0	0,99	0,99	5	2
Повідомити оператору про закриття інциденту	1 - Робоча	0,99	0	0	3	1

Рисунок 4.17 – Довідник операцій

Після того як було введено ФМ (рис 4.18), треба виконати її редукцію, для цього в головному меню необхідно обрати “Автоматична редукція мережі”, далі ввести номери комірок з початком і кінцем опису. Підсистема автоматично знайде типові функціональні структури в ФМ і замінить їх на еквівалентні типові одиниці роблячи при цьому відповідні розрахунки. На рисунках 4.19-4.21 зображено всі кроки редукції ФМ, доки не залишилась єдина операція. Показники часу і надійності якої і є показниками для виконання всього алгоритму.

Номер ліп	Назва ТФО	Код ТФО	Позначення ТФО	Продовження								Згортання		Імовірність виконання операції без помилок	Імовірність того, що фактично виконане буде визнане правильним	Імовірність того, що фактично виконане буде визнане неправильним	Мат. очікувані в часу виконання операції	Дисперсія часу виконання	Операція	
				При виконанні умови	При невиконанні КФ	При невиконанні КР	На провадженні цикла	Номер за вихід з цикла	Кількість повторів в циклі	Код ТФС для редукції	Номер 1-й ТФО для редукції	Позначення ТФО на наступному етапі	$B_1^{1'}$							$K_1^{11}$
№01	№01ТФЕ	№04	№07FE	№02	№03	№05	№06	№07	№08	№01	№04	ТХ05								
1.01 - Стартер	91 S		2	0	0	0	0	0	0				0,000	0,000	0,000	0,000	0,000	0,000		
2.1 - Робоча	1 P1		3	0	0	0	0	0	0				0,98	0	0	5	2		Отримати інформацію про інцидент по рації	
3.1 - Робоча	1 P2		4	0	0	0	0	0	0				0,99	0	0	3	1		Повідомити оператору про успішне отримання інформації про інцидент	
4.1 - Робоча	1 P3		5	0	0	0	0	0	0				0,985	0	0	180	60		Приїхати до місця інциденту	
5.1 - Робоча	1 P4		6	0	0	0	0	0	0				0,97	0	0	5	2		Повідомити оператору про прибуття	
6.1 - Робоча	1 P5		7	0	0	0	0	0	0				0,965	0	0	150	30		Оглянути об'єкт ззовні	
7.1 - Робоча	1 P6		8	0	0	0	0	0	0				0,97	0	0	5	2		Повідомити оператору про результати огляду об'єкту	
8.1 - Робоча	1 P7		9	0	0	0	0	0	0				0,97	0	0	150	30		Оглянути об'єкт всередині	
9.1 - Робоча	1 P8		10	0	0	0	0	0	0				0,97	0	0	5	2		Повідомити оператору про результати огляду об'єкту	
10.1 - Робоча	1 P9		11	0	0	0	0	0	0				0,965	0	0	60	15		Виявити причину інциденту	
3 - Контроль функціонування	3 K1		12	10	0	0	0	0	0				0	0,965	0,965	10	3		Контроль правильності виявлення причини інциденту	
12.1 - Робоча	1 P10		13	0	0	0	0	0	0				0,96	0	0	70	30		Проведення заходів оперативного реагування	
13.1 - Робоча	1 P11		14	0	0	0	0	0	0				0,97	0	0	5	2		Повідомити оператору причину виникнення інциденту	
14.1 - Робоча	1 P12		15	0	0	0	0	0	0				0,96	0	0	30	10		Перезавантажити пристрій спостереження	
3 - Контроль функціонування	3 K2		16	14	0	0	0	0	0				0	0,98	0,98	5	2		Контроль над тим що пристрій перезавантажено	
16.1 - Робоча	1 P13		17	0	0	0	0	0	0				0,97	0	0	5	2		Повідомити оператору при перезавантаженні пристрою спостереження	
17.1 - Робоча	1 P14		18	0	0	0	0	0	0				0,99	0	0	40	15		Перезачинити об'єкт	
3 - Контроль функціонування	3 K3		19	17	0	0	0	0	0				0	0,99	0,99	5	2		Контроль над тим що об'єкт зачинено	
19.1 - Робоча	1 P15		20	0	0	0	0	0	0				0,99	0	0	3	1		Повідомити оператору про закриття інциденту	
20.99 - Фіншер	99 F		21	0	0	0	0	0	0				0,000	0,000	0,000	0,000	0,000			

Рисунок 4.18 – Результат введення опису ФМ

Номер п/п	Назва ТФО	Код ТФО	Позначення ТФО	Продовження					Згорання			Імовірність виконання робочої операції без помилок	Імовірність того, що фактично виконання буде визнане правильним	Імовірність того, що фактично виконання буде визнане неправильним	Мат. очікуванн я часу виконання операції	Дисперсія часу виконання	
				При виконанні умови	При невиконанні КФ	При невиконанні КР	На продовження цикла	Номер на вихід з цикла	Кількість повторів в циклі	Код ТФС для редукції	Номер 1-й ТФО для редукції						Позначення ТФО на наступному етапі
N01	NameTFO	K04	ObTFO	N02	N03	N05	N06	N07	N08	K01	N04	TX05	B11	K111	K100	M(T)	D(T)
1	91-Стартер	91	S	2	0	0	0	0	0				0,000	0,000	0,000	0,000	0,000
2	1 - Робоча	1	P1	3	0	0	0	0	0	RR	2	Pa1	0,98	0	0	5	2
3	1 - Робоча	1	P2	4	0	0	0	0	0	RR	2	Pa1	0,98	0	0	3	1
4	1 - Робоча	1	P3	5	0	0	0	0	0	RR	2	Pa1	0,985	0	0	180	60
5	1 - Робоча	1	P4	6	0	0	0	0	0	RR	2	Pa1	0,97	0	0	5	2
6	1 - Робоча	1	P5	7	0	0	0	0	0	RR	2	Pa1	0,965	0	0	150	40
7	1 - Робоча	1	P6	8	0	0	0	0	0	RR	2	Pa1	0,97	0	0	5	2
8	1 - Робоча	1	P7	9	0	0	0	0	0	RR	2	Pa1	0,97	0	0	150	40
9	1 - Робоча	1	P8	10	0	0	0	0	0	RR	2	Pa1	0,97	0	0	5	2
10	1 - Робоча	1	P9	11	0	0	0	0	0				0,985	0	0	60	15
3	Контроль функціонування	3	K1	12	10	0	0	0	0				0	0,965	0,965	10	3
12	1 - Робоча	1	P10	13	0	0	0	0	0	RR	12	Pa2	0,98	0	0	70	30
13	1 - Робоча	1	P11	14	0	0	0	0	0	RR	12	Pa2	0,97	0	0	5	2
14	1 - Робоча	1	P12	15	0	0	0	0	0				0,96	0	0	30	10
3	Контроль функціонування	3	K2	16	14	0	0	0	0				0	0,96	0,96	5	2
16	1 - Робоча	1	P13	17	0	0	0	0	0				0,97	0	0	5	2
17	1 - Робоча	1	P14	18	0	0	0	0	0				0,99	0	0	40	15
3	Контроль функціонування	3	K3	19	17	0	0	0	0				0	0,99	0,99	5	2
19	1 - Робоча	1	P15	20	0	0	0	0	0				0,99	0	0	3	1
20	99-Фінішер	99	F	21	0	0	0	0	0				0,000	0,000	0,000	0,000	0,000

Рисунок 4.19 – Перший крок редукції ФМ

Номер п/п	Назва ТФО	Код ТФО	Позначення ТФО	Продовження					Згорання			Імовірність виконання робочої операції без помилок	Імовірність того, що фактично виконання буде визнане правильним	Імовірність того, що фактично виконання буде визнане неправильним	Мат. очікуванн я часу виконання операції	Дисперсія часу виконання	
				При виконанні умови	При невиконанні КФ	При невиконанні КР	На продовження цикла	Номер на вихід з цикла	Кількість повторів в циклі	Код ТФС для редукції	Номер 1-й ТФО для редукції						Позначення ТФО на наступному етапі
N01	NameTFO	K04	ObTFO	N02	N03	N05	N06	N07	N08	K01	N04	TX05	B11	K111	K100	M(T)	D(T)
1	91-Стартер	91	S	2	0	0	0	0	0				0	0	0	0	0
2	1 - Робоча	1	P1	3	0	0	0	0	0				0,816416456	0	0	503	149
3	1 - Робоча	1	P9	4	0	0	0	0	0	RK	3	Pa3	0,985	0	0	60	15
3	Контроль функціонування	3	K1	5	3	0	0	0	0	RK	3	Pa3	0	0,985	0,964999974	10	3
5	1 - Робоча	1	P2	6	0	0	0	0	0				0,950000	0,000000	0,000000	75,000000	32,000000
6	1 - Робоча	1	P12	7	0	0	0	0	0	RK	6	Pa4	0,960000	0,000000	0,000000	30,000000	10,000000
7	3 - Контроль функціонування	3	K2	8	6	0	0	0	0	RK	6	Pa4	0,000000	0,980000	0,980000	5,000000	2,000000
8	1 - Робоча	1	P13	9	0	0	0	0	0				0,970000	0	0	5,000000	2,000000
9	1 - Робоча	1	P14	10	0	0	0	0	0	RK	9	Pa5	0,99	0,000000	0,000000	40,000000	15,000000
3	Контроль функціонування	3	K3	11	9	0	0	0	0	RK	9	Pa5	0	0,99	0,99000001	5	2
11	1 - Робоча	1	P15	12	0	0	0	0	0				0,99	0,000000	0,000000	3,000000	1,000000
12	99-Фінішер	99	F	21	0	0	0	0	0				0,000000	0	0	0,000000	0,000000

Рисунок 4.20 – Другий крок редукції ФМ

Номер п/п	Назва ТФО	Код ТФО	Позначення ТФО	Продовження					Згорання			Імовірність виконання робочої операції без помилок	Імовірність того, що фактично виконання буде визнане правильним	Імовірність того, що фактично виконання буде визнане неправильним	Мат. очікуванн я часу виконання операції	Дисперсія часу виконання	
				При виконанні умови	При невиконанні КФ	При невиконанні КР	На продовження цикла	Номер на вихід з цикла	Кількість повторів в циклі	Код ТФС для редукції	Номер 1-й ТФО для редукції						Позначення ТФО на наступному етапі
N01	NameTFO	K04	ObTFO	N02	N03	N05	N06	N07	N08	K01	N04	TX05	B11	K111	K100	M(T)	D(T)
1	91-Стартер	91	S	2	0	0	0	0	0				0	0	0	0	0
2	1 - Робоча	1	P1	3	0	0	0	0	0	RR	2	Pa6	0,816416456	0	0	503	149
3	1 - Робоча	1	P3	4	0	0	0	0	0	RR	2	Pa6	0,99447978	0	0	73,60285997	284,1072388
4	1 - Робоча	1	P2	5	0	0	0	0	0	RR	2	Pa6	0,9506	0	0	75	32
5	1 - Робоча	1	P4	6	0	0	0	0	0	RR	2	Pa6	0,999180383	0	0	37,17077318	93,43358612
6	1 - Робоча	1	P13	7	0	0	0	0	0	RR	2	Pa6	0,97	0	0	5	2
7	1 - Робоча	1	P5	8	0	0	0	0	0	RR	2	Pa6	0,99989798	0	0	45,90899817	59,07459641
8	1 - Робоча	1	P15	9	0	0	0	0	0	RR	2	Pa6	0,99	0	0	3	1
9	99-Фінішер	99	F	21	0	0	0	0	0				0	0	0	0	0

Номер п/п	Назва ТФО	Код ТФО	Позначення ТФО	Продовження					Згорання			Імовірність виконання робочої операції без помилок	Імовірність того, що фактично виконання буде визнане правильним	Імовірність того, що фактично виконання буде визнане неправильним	Мат. очікуванн я часу виконання операції	Дисперсія часу виконання	
				При виконанні умови	При невиконанні КФ	При невиконанні КР	На продовження цикла	Номер на вихід з цикла	Кількість повторів в циклі	Код ТФС для редукції	Номер 1-й ТФО для редукції						Позначення ТФО на наступному етапі
N01	NameTFO	K04	ObTFO	N02	N03	N05	N06	N07	N08	K01	N04	TX05	B11	K111	K100	M(T)	D(T)
1	91-Стартер	91	S	2	0	0	0	0	0				0	0	0	0	0
2	1 - Робоча	1	Pa6	3	0	0	0	0	0				0,744154709	0	0	742,6826313	620,6154175
3	99-Фінішер	99	F	21	0	0	0	0	0				0	0	0	0	0

Рисунок 4.21 – Останні кроки редукції ФМ



Таким чином було виконано оцінку часу та надійності виконання алгоритмів діяльності шістьох бригад для кожного з чотирьох рівнів загрози. В таблиці 4.2 наведені результати оцінювання алгоритмів діяльності бригад для всіх рівнів загрози.

Таблиця 4.2 – Результати оцінювання алгоритмів діяльності бригад

Бригада	Рівень загрози інциденту	Ймовірність правильного виконання	Мат. очікування часу виконання	Дисперсія часу виконання
1	Наднизький	0,677	703,0	618,8
1	Низький	0,663	704,5	736,9
1	Середній	0,648	706,1	864,5
1	Високий	0,633	707,7	1002,6
2	Наднизький	0,670	704,1	706,5
2	Низький	0,666	705,3	799,5
2	Середній	0,655	706,0	864,5
2	Високий	0,641	706,8	932,1
3	Наднизький	0,702	774,9	787,8
3	Низький	0,694	775,6	851,6
3	Середній	0,687	776,4	918,0
3	Високий	0,679	798,4	1183,0
4	Наднизький	0,721	743,7	686,6
4	Низький	0,705	744,5	746,8
4	Середній	0,690	745,2	809,4
4	Високий	0,675	746,0	874,5
5	Наднизький	0,744	742,7	620,6
5	Низький	0,732	743,4	677,4
5	Середній	0,720	744,1	736,5
5	Високий	0,709	744,9	797,8
6	Наднизький	0,684	702,6	618,5
6	Низький	0,684	704,5	736,9
6	Середній	0,669	705,6	831,6
6	Високий	0,654	706,8	932,1

## 4.5 Реалізація підсистеми вибору оптимального варіанту розподілу інцидентів

Основною проблемою оператора є вибір бригади, якій доцільно доручити завдання по усуненню інциденту. На практиці можуть виникнути різноманітні постановки задач по оптимізації, тому було поставлено завдання розробити сукупність моделей з яких при необхідності оператор може обрати ту, яка відповідає поточній ситуації. Реалізована підсистема дає можливість обирати оптимальну бригаду вирішуючи різні задачі (рис 4.22), як однокритеріальні, так і багатокритеріальні.

Однокритеріальні задачі включають в себе можливість швидко обрати найближчу бригаду (рис 4.23) або бригаду з найкращим показником правильного виконання алгоритму (рис 4.24).

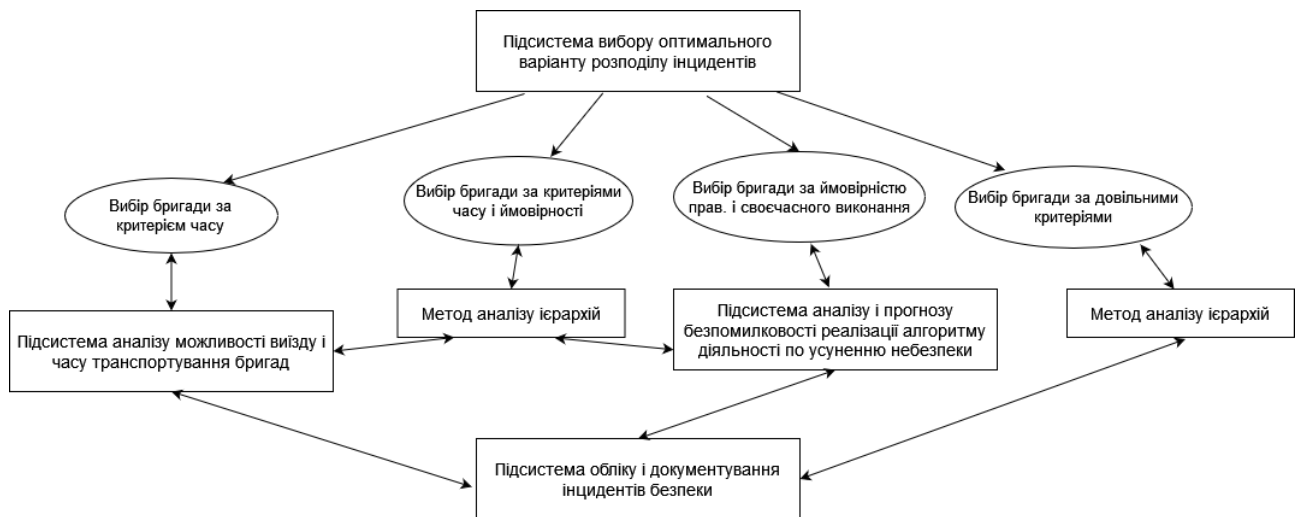


Рисунок 4.22 – Структура підсистеми вибору оптимального варіанту розподілу інцидентів

Поточний статус: Взято до опрацювання й направлено бригаду  
 Бригада: 6 - Венглинський Олег Валентинович, Мороз Олександр Макарович  
 Змінити статус: 1 - Взято до опрацювання й направлено бригаду  
 Обрати бригаду: 2 - Філатов Арсеній Андрійович, Бенедюк Артем Ігорович  
 Додати

Обрати найближчу бригаду  
 Обрати бригаду з кращим показником ймовірності своєчасного та правильного виконання  
 Обрати оптимальну бригаду враховуючи обидва критерії  
 Обрати оптимальну бригаду за довільними критеріями

Бригада: 2, Час: 138с, Відстань: 3046м, Ймовірність: 66%  
 Бригада: 5, Час: 150с, Відстань: 3296м, Ймовірність: 72%  
 Бригада: 4, Час: 168с, Відстань: 3686м, Ймовірність: 69%  
 Бригада: 1, Час: 245с, Відстань: 5387м, Ймовірність: 65%  
 Бригада: 6, Час: 257с, Відстань: 5664м, Ймовірність: 67%  
 Бригада: 3, Час: 365с, Відстань: 8030м, Ймовірність: 69%

Рисунок 4.23 – Вибір оптимальної бригади за критерієм часу прибуття до об'єкту

Поточний статус: Взято до опрацювання й направлено бригаду  
 Бригада: 6 - Венглинський Олег Валентинович, Мороз Олександр Макарович  
 Змінити статус: 1 - Взято до опрацювання й направлено бригаду  
 Обрати бригаду: 5 - Горбач Ігор Віталійович, Федоров Сергій Русланович  
 Додати

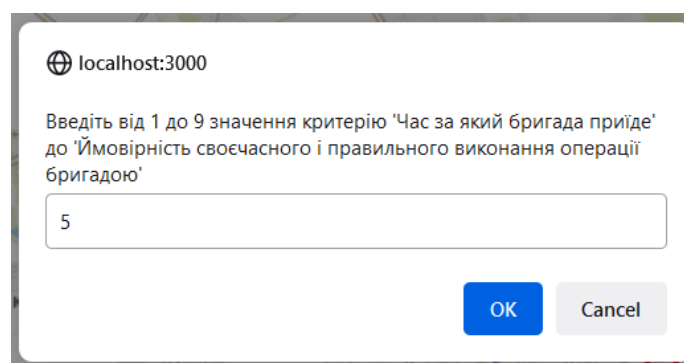
Обрати найближчу бригаду  
 Обрати бригаду з кращим показником ймовірності своєчасного та правильного виконання  
 Обрати оптимальну бригаду враховуючи обидва критерії  
 Обрати оптимальну бригаду за довільними критеріями

Бригада: 5, Ймовірність: 72%, Час: 150с, Відстань: 3296м  
 Бригада: 4, Ймовірність: 69%, Час: 168с, Відстань: 3686м  
 Бригада: 3, Ймовірність: 69%, Час: 365с, Відстань: 8030м  
 Бригада: 6, Ймовірність: 67%, Час: 257с, Відстань: 5664м  
 Бригада: 2, Ймовірність: 66%, Час: 138с, Відстань: 3046м  
 Бригада: 1, Ймовірність: 65%, Час: 245с, Відстань: 5387м

Рисунок 4.24 – Вибір оптимальної бригади за критерієм ймовірності правильного виконання алгоритму діяльності по усуненню інциденту

Багатокритеріальні задачі вибору оптимальної бригади вирішуються за допомогою інтегрованого в підсистему методу аналізу ієрархій. Є декілька варіантів постановки багатокритеріальної задачі, обрати оптимальну бригаду за критеріями часу прибуття до об'єкту і правильності виконання алгоритму діяльності, або ж знайти оптимальну бригаду за власними, довільними критеріями, тобто в залежності від ситуації, що виникла. В першому варіанті в якості вхідних даних використовуються результати розрахунків інших підсистеми та експертні оцінки важливості критеріїв, які необхідно ввести оператору (рис 4.25) та отримуються результати пріоритетності вибору бригад (рис 4.26).

При вирішенні задачі за довільними критеріями, оператору потрібно ввести відповідно критерії (рис 4.27), та результати попарних порівнянь важливості критеріїв (рис 4.28) і окремо альтернатив відносно кожного критерію, в результаті отримується діаграма, що показує пріоритет вибору кожної бригади. На рисунках 4.29-4.30 зображено приклад вирішення задачі по вибору оптимальної бригади за критеріями: відстань до об'єкту, ефективність виконання діяльності по усуненню інциденту, важливість об'єкту та рівень загрози інциденту.



The image shows a web browser window with a dialog box. The address bar shows 'localhost:3000'. The dialog box contains the following text: 'Введіть від 1 до 9 значення критерію "Час за який бригада приїде" до "Ймовірність своєчасного і правильного виконання операції бригадою"'. Below the text is a text input field containing the number '5'. At the bottom right of the dialog box are two buttons: 'OK' (blue) and 'Cancel' (grey).

Рисунок 4.25 – Введення оцінки важливості критеріїв

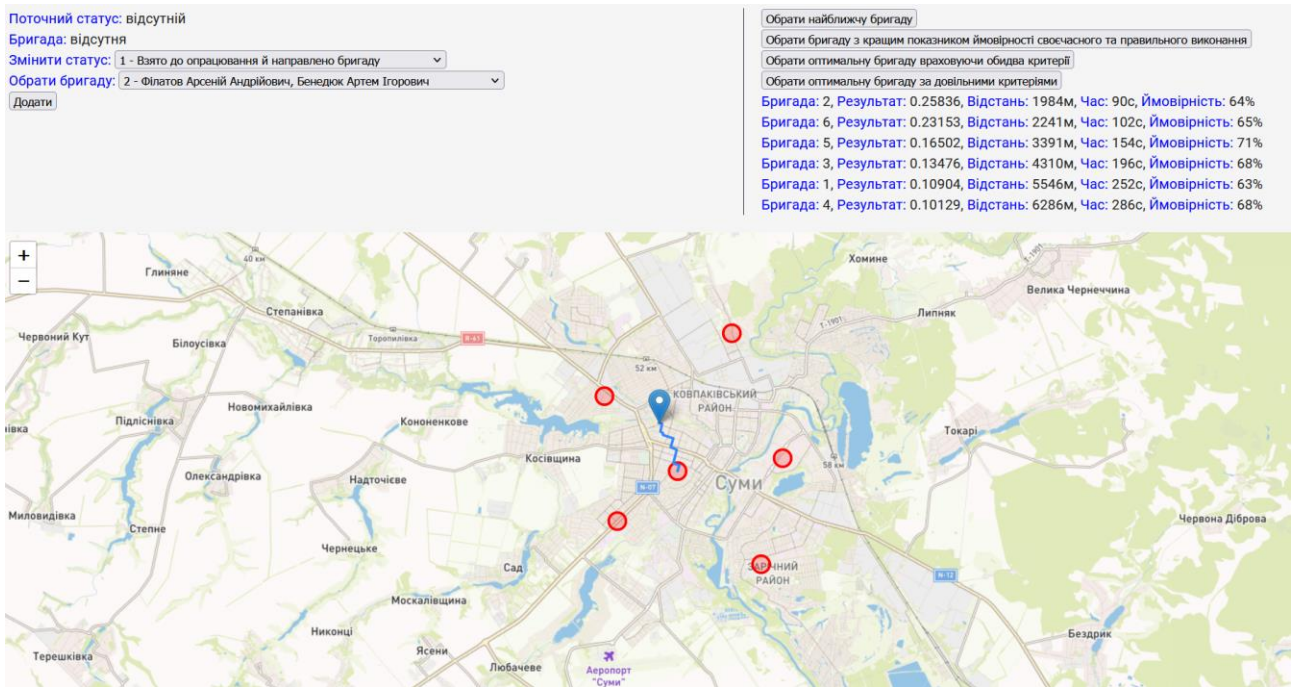


Рисунок 4.26 – Вибір оптимальної бригади за критеріями часу прибуття та ймовірності правильного виконання

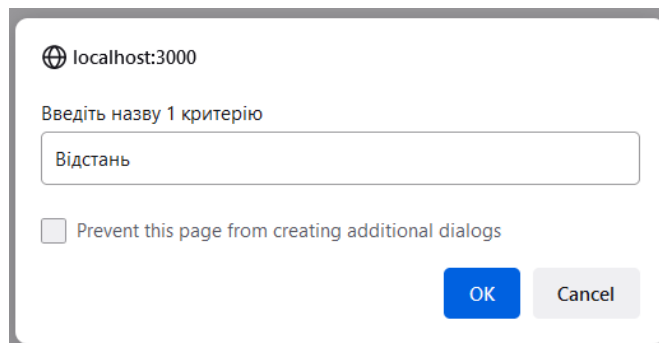


Рисунок 4.27 – Введення критеріїв

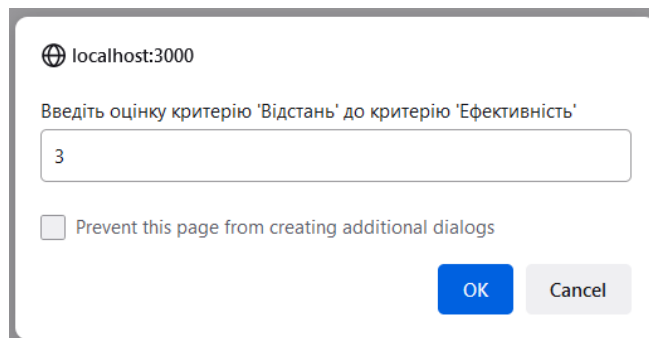


Рисунок 4.28 – Попарне порівняння критеріїв

```

rankingMatrix: (Higher score is better)
-----
|           | Бригада 1 | Бригада 2 | Бригада 3 | Бригада 4 | Бригада 5 | Бригада 6 |
|-----|-----|-----|-----|-----|-----|-----|
| Відстань | 0.259 | 0.061 | 0.198 | 0.199 | 0.057 | 0.227 |
| Ефективність виконання | 0.185 | 0.310 | 0.038 | 0.143 | 0.196 | 0.127 |
| Важливість | 0.266 | 0.074 | 0.283 | 0.083 | 0.199 | 0.095 |
| Загроза | 0.119 | 0.185 | 0.201 | 0.296 | 0.175 | 0.025 |
-----

ranked item scores: (Higher score is better)
-----
|      | Score |
|-----|-----|
| Бригада 1 | 0.201 |
| Бригада 2 | 0.129 |
| Бригада 3 | 0.227 |
| Бригада 4 | 0.189 |
| Бригада 5 | 0.171 |
| Бригада 6 | 0.084 |
-----

```

Рисунок 4.29 – Вирішення багатокритеріальної задачі за довільними критеріями

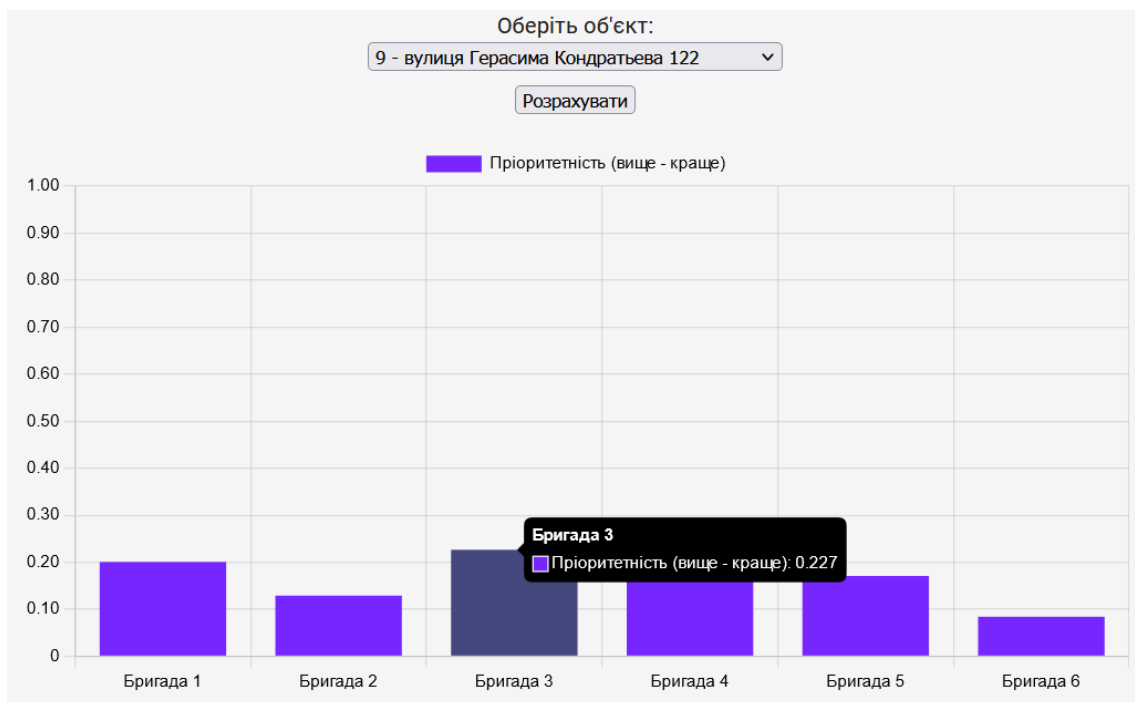


Рисунок 4.30 – Діаграма з результатами пріоритетності вибору бригад

## 4.6 Реалізація підсистеми обліку і документування інцидентів безпеки

Для того щоб забезпечити облік й документування робочого процесу інцидентів прийшлося теж розробити окрему підсистему, вона є необхідною лише для того щоб забезпечити інформаційне середовище оператора. При виникненні нового інциденту, в лівому нижньому куті з'являється індикатор (рис 4.31), який не зникне, доки оператор не зайде на сторінку інциденту, після цього вважається що інцидент був помічений оператором і роботу над його вирішенням розпочато. На сторінці інциденту, після того як оператор обрав оптимальну бригаду, йому необхідно її призначити для початку роботи над вирішенням інциденту, далі по мірі виконання операцій, змінювати поточний статус інциденту поки його не буде вирішено (рис. 4.32). Можливі статуси інцидентів зберігаються в базі даних, в таблиці statuses (рис. 4.33), якщо статус інциденту обрано як “Операцію виконано” – він закривається і зникає зі сторінки активних інцидентів.

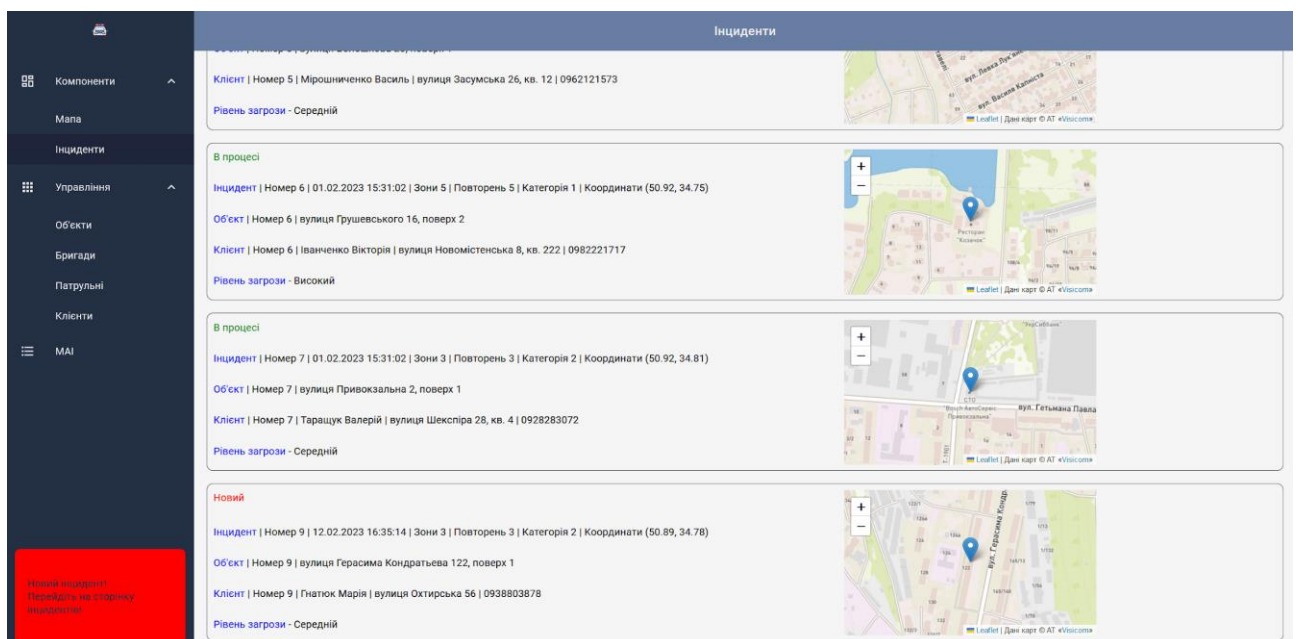


Рисунок 4.31 – Індикатор появи нового інциденту

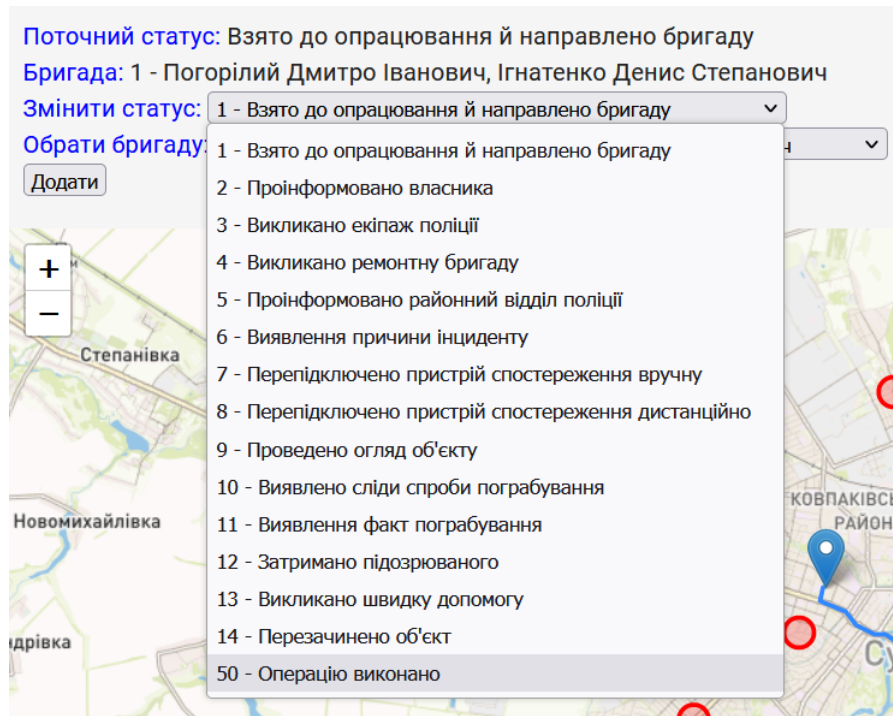


Рисунок 4.32 – Можливі статуси інциденту

```
1 • SELECT * FROM operator_helper.statuses;
```

id	title	description
1	Взято до опрацювання й направлено бригаду	Операцію взято до уваги, направлено бригаду і розпочато дії щодо її виконання
2	Проінформовано власника	NULL
3	Викликано екіпаж поліції	NULL
4	Викликано ремонтну бригаду	NULL
5	Проінформовано районний відділ поліції	NULL
6	Виявлення причини інциденту	NULL
7	Перепідключено пристрій спостереження вручну	NULL
8	Перепідключено пристрій спостереження дистанційно	NULL
9	Проведено огляд об'єкту	NULL
10	Виявлено сліди спроби пограбування	NULL
11	Виявлення факт пограбування	NULL
12	Затримано підозрюваного	NULL
13	Викликано швидку допомогу	NULL
14	Перезачинено об'єкт	NULL
50	Операцію виконано	Виконання операції було закінчено, інцидент вважається закритим
NULL	NULL	NULL

Рисунок 4.33 – Таблиця statuses в базі даних



Для того щоб відслідкувати як виконувалось вирішення інциденту, яка бригада працювала над усуненням, в який час виконувались ті чи інші операції, тощо - можна використати базу даних написавши відповідні запити. На рисунку 4.34 зображено приклад реалізації запиту до таблиці operations для отримання інформації про інцидент з номером 6, на рисунку 4.35 зображено результат запиту.

```
use operator_helpers;
SELECT incident_id, id, (SELECT patrolmen.full_name from patrolmen where patrolmen.id=(select patrols.first_patrolman_id from patrols where patrols.id=operations.patrol_id)) as first_patrolman,
(SELECT patrolmen.full_name from patrolmen where patrolmen.id=(SELECT patrols.second_patrolman_id from patrols where patrols.id=operations.patrol_id)) as second_patrolman,
(SELECT statuses.title from statuses where statuses.id=operations.status_id) as status, (SELECT timestamps.time from timestamps where timestamps.id=operations.timestamp_id) as time,
(SELECT timestamps.date from timestamps where timestamps.id=operations.timestamp_id) as date
FROM operations WHERE incident_id=6 ORDER BY timestamp_id
```

Рисунок 4.34 – Запит до таблиці operations

incident_id	id	first_patrolman	second_patrolman	status	time	date
6	10	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Взято до опрацювання й направлено бригаду	21:02:02	2023-04-24
6	11	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Проінформовано власника	21:02:08	2023-04-24
6	12	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Викликано екіпаж поліції	21:02:11	2023-04-24
6	13	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Виявлення причини інциденту	21:02:18	2023-04-24
6	14	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Викликано ремонтну бригаду	21:02:20	2023-04-24
6	15	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Перепідключено пристрій спостереження вручну	21:02:24	2023-04-24
6	16	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Проведено огляд об'єкту	21:02:49	2023-04-24
6	17	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Перезачинено об'єкт	21:04:01	2023-04-24
6	18	Філатов Арсеній Андрійович	Бенедюк Артем Ігорович	Операцію виконано	21:04:04	2023-04-24

Рисунок 4.35 – Результат запиту

## ВИСНОВКИ

З появою нових інформаційно-комунікаційних технологій все більшої популярності набувають процеси розвитку міст за концепцією смарт-сіті. Запровадження таких технологій в роботі оператора централізованого пульта спостереження за складними об'єктами покращує ефективність його роботи та зменшує час на прийняття рішень, тим самим підвищуючи безпеку й благоустрій міста.

Інформаційна технологія підтримки прийняття рішень при управлінні безпекою складних об'єктів забезпечує прийняття рішень на основі об'єктивних кількісних показників та передбачає зменшення напруженості, а також когнітивний комфорт для операторів-керівників і сприяє підвищенню безпеки об'єктів, що охороняються.

Наукова новизна полягає в тому, що на відміну від існуючих моделей інформаційного забезпечення діяльності операторів пульта спостереження, орієнтованих на інтуїтивні методи прийняття рішень, розроблені моделі використовують формальний апарат опису діяльності по усуненню інцидентів на основі методології функціональних мереж та апарат ієрархічної декомпозиції проблеми, який дозволяє приймати рішення на основі багатокритеріального оцінювання альтернативних сценаріїв усунення інцидентів.

Практичне значення полягає в тому, що використання інформаційної технології підтримки діяльності оператора пульта спостереження за складними об'єктами забезпечує швидкий моніторинг інцидентів, місця знаходження бригад, вибір оптимальної бригади в залежності від обраних критеріїв, облік й керування робочим процесом при вирішенні інциденту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рудик В.В., Лавров Є.А., Пасько Н.Б. Моделювання діяльності людини-оператора централізованого пульта спостереження за об'єктами // Інформатика, Математика, Автоматика (ІМА - 2021) Матеріали та програма міжнародної науково-технічної конференції студентів та молодих учених, Суми–Нур-Султан, 19–23 квітня 2021 р , Суми, СумДУ, 2021, С. 109-110

2. Рудик В.В., Лавров Є.А. Класифікатор ситуацій для підтримки прийняття рішень операторів централізованих пультів спостереження за об'єктами// Інформатика, Математика, Автоматика (ІМА - 2021) Матеріали та програма міжнародної науково-технічної конференції студентів та молодих учених, Суми–Нур-Султан, 19–23 квітня 2021 р , Суми, СумДУ , 2021, С. 107-108

3. Рудик В.В., Лавров Є.А. Інформаційна технологія підтримки прийняття рішень при управлінні безпекою складних об'єктів // Інформатика, Математика, Автоматика (ІМА - 2023) Матеріали та програма міжнародної науково-технічної конференції студентів та молодих учених, Суми–Астана, 24-28 квітня 2023р, Суми, СумДУ, 2023, С. 184-185

4. Рудик В.В., Лавров Є.А. Метод визначення рівня загрози для підтримки прийняття рішень операторів систем управління безпекою складних об'єктів // Інформатика, Математика, Автоматика (ІМА - 2023) Матеріали та програма міжнародної науково-технічної конференції студентів та молодих учених, Суми–Астана, 24-28 квітня 2023р, Суми, СумДУ, 2023, С. 186-187

5. Лавров Є.А., Рудик В.В., Клименко І.В, Солонар І.О., Чечель О.В. Людино-системний підхід до вирішення задач забезпечення надійності обробки інформації в системах автоматизованого проектування і управління // Інформатика, Математика, Автоматика (ІМА - 2023) Матеріали та програма

міжнародної науково-технічної конференції студентів та молодих учених, Суми–Астана, 24-28 квітня 2023р, Суми, СумДУ, 2023, С. 176-177

6. Рудик, В.В. Моделювання діяльності людини-оператора централізованого пульта спостереження за об'єктами [Текст]: робота на здобуття кваліфікаційного рівня бакалавра; спец.: 122 - комп'ютерні науки / В.В. Рудик; наук. кер. Є.А. Лавров. – Суми: СумДУ, 2021. – 121 с.

7. Організація роботи централізованої охорони [Електронний ресурс] – Режим доступу до ресурсу: <http://parusconsultant.com/?doc=07Z5925E93>.

8. Monitoring Station [Електронний ресурс] – Режим доступу до ресурсу: <https://www.agmonitoring.com/resources/what-is-central-station-monitoring-pillar>.

9. Police Control Room [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kesinoconsoles.com/info/how-does-a-police-control-room-work-i00035i1.html>.

10. AI video surveillance [Електронний ресурс] – Режим доступу до ресурсу: <https://www.scylla.ai/advantages-of-using-artificial-intelligence-in-video-surveillance/>.

11. Drones for security [Електронний ресурс] - Режим доступу до ресурсу: <https://www.deltadrone.com/en/how-does-the-drone-allow-for-surveillance/>.

12. Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, 22, 3–21.

13. Anthopoulos, L. (2019). *Smart city emergence: Cases from around the world*. Elsevier.

14. Silva, B. N., Khan, M., & Han, K. (2018). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable Cities and Society*, 38, 697–713.

15. Kashef, M., Visvizi, A., & Troisi, O. (2021). Smart city as a smart service system: Human-computer interaction and smart city surveillance systems. *Computers in Human Behavior*, 124. <https://doi.org/10.1016/j.chb.2021.106923>.

16. Ben Mabrouk, A., & Zagrouba, E. (2018). Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2017.09.029>
17. Maciel, H. S., Cardoso, I., Ramos, H. S., Rodrigues, J. J. P. C., & Aquino, A. L. L. (2017). An integrated access control and lighting configuration system for smart buildings. *Journal of Communications Software and Systems*. <https://doi.org/10.24138/jcomss.v13i2.378>
18. Sahoo, K. C., & Pati, U. C. (2017). IoT based intrusion detection system using PIR sensor. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*.
19. Abbas, A. H., Habelalmateen, M. I., Jurdi, S., Audah, L., & Alduais, N. A. M. (2019). GPS based location monitoring system with geo-fencing capabilities. *Advances in electrical and electronic engineering: from theory to applications (series 2): Proceedings of the International Conference of Electrical and Electronic Engineering*.
20. Sahani, M., Subudhi, S., & Mohanty, M. N. (2016). Design of face recognition based embedded home security system. *KSII Transactions on Internet and Information Systems*. <https://doi.org/10.3837/tiis.2016.04.016>
21. Geotab [Электронный ресурс] – Режим доступа до ресурсу: <https://www.geotab.com/industries/police-first-responders-fleet/>
22. Webfleet Solutions [Электронный ресурс] – Режим доступа до ресурсу: [https://www.webfleet.com/en\\_gb/webfleet/](https://www.webfleet.com/en_gb/webfleet/)
23. Fleetio [Электронный ресурс] – Режим доступа до ресурсу: <https://www.fleetio.com/features/>
24. Wialon [Электронный ресурс] – Режим доступа до ресурсу: <https://gurtam.com/en/wialon/online-monitoring>
25. Berrar, D. (2019). Bayes' Theorem and Naive Bayes Classifier. In *Encyclopedia of Bioinformatics and Computational Biology* (pp. 403–412). Elsevier.

26. Song, Y.-Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2), 130–135. <https://doi.org/10.11919/j.issn.1002-0829.215044>
27. Kriegeskorte, N., & Golan, T. (2019). Neural network models and deep learning. *Current Biology: CB*, 29(7), R231–R236.
28. Panchal, S., & Shrivastava, A. K. (2022). Landslide hazard assessment using analytic hierarchy process (AHP): A case study of National Highway 5 in India. *Ain Shams Engineering Journal*, 13(3), 101626. <https://doi.org/10.1016/j.asej.2021.10.021>
29. Ткачова О. К. (2015) Метод Сааті при прийнятті управлінських рішень. *Держава та регіони № 4*. - С. 92-96.
30. Pant, S., Kumar, A., Ram, M., Klochkov, Y., & Sharma, H. K. (2022). Consistency indices in analytic hierarchy process: A review. *Mathematics*, 10(8), 1206. <https://doi.org/10.3390/math10081206>
31. Lavrov, E., Pasko, N., Siryk, O., Burov, O., & Osadchyi, V. (2020). Ergonomics of cyberspace. mathematical modeling to create groups of operators for error-free and timely implementation of functions in a distributed control system. In *CEUR Workshop Proceedings (Vol. 2740, pp. 380–385)*. CEUR-WS.
32. Lavrov, E., Paderno, P., Burkov, E., Volosiuk, A., & Lung, V. D. (2020). Expert assessment systems to support decision-making for sustainable development of complex technological and socioeconomic facilities. In *E3S Web of Conferences (Vol. 166)*. EDP Sciences. <https://doi.org/10.1051/e3sconf/202016611002>
33. Lavrov, E., Pasko, N., & Siryk, O. (2020). Information technology for assessing the operators working environment as an element of the ensuring automated systems ergonomics and reliability. In *Proceedings - 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020 (pp. 570–575)*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/TCSET49122.2020.235497>
34. Lavrov, E. A., Paderno, P. I., Burkov, E. A., Siryk, O. E., & Pasko, N. B. (2020). Information Technology for Modeling Human-machine Control Systems and

Approach to Integration of Mathematical Models for Its Improvement. In Proceedings of 2020 23rd International Conference on Soft Computing and Measurements, SCM 2020 (pp. 117–120). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/SCM50615.2020.9198791>

35. Lavrov, E., et al. The Methodology of Managed Functional Networks for Organizing Effective and Adaptive Human-Machine Dialogue in Automated Systems 2021 CEUR Workshop Proceedings 2021 3013, p. 428-437

36. Lavrov E. et al. (2021) Analysis of Working Conditions and Modeling of Activity Algorithms for Contact-Center Operators. In: Sumpor D., Jambrošić K., Jurčević Lulić T., Milčić D., Salopek Čubrić I., Šabarić I. (eds) Proceedings of the 8th International Ergonomics Conference. ERGONOMICS 2020. Advances in Intelligent Systems and Computing, vol 1313. Springer, Cham. [https://doi.org/10.1007/978-3-030-66937-9\\_14](https://doi.org/10.1007/978-3-030-66937-9_14)

37. Lavrov, E., Pasko, N., Siryk, O., Burov, O., & Natalia, M. (2020). Mathematical Models for Reducing Functional Networks to Ensure the Reliability and Cybersecurity of Ergatic Control Systems. In Proceedings - 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020 (pp. 179–184). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/TCSET49122.2020.235418>

38. Lavrov, E., Pasko, N., Siryk, O., Mukoseev, V., & Dubovyk, S. (2020). Automation of reliability assessment of functional elements of flexible automated production based on functional network methodology. In CEUR Workshop Proceedings (Vol. 2740, pp. 357–364). CEUR-WS.

39. E. Lavrov, N. Pasko, O. Siryk, P. Paderno and E. Burkov, "Models and Information Technology for Reliable Design of the Functioning Processes of Flexible Production Systems as Complex Human-Machine Systems," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020, pp. 673-678, doi: 10.1109/PICST51311.2020.9467894.

40. Lavrov, E., Lavrova, O., Pasko, N., Kyzenko, V., & Savina, N. (2019). Assessment of the reliability of a human operator in access systems to information resources. In 2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019 - Proceedings (pp. 51–56). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/PICST47496.2019.9061495>

41. Tserng, H.-P., Cho, I.-C., Chen, C.-H., & Liu, Y.-F. (2021). Developing a risk management process for infrastructure projects using IDEF0. *Sustainability*, 13(12), 6958. <https://doi.org/10.3390/su13126958>

42. Chren, S., Buhnova, B., Macak, M., Daubner, L., & Rossi, B. (2019). Mistakes in UML diagrams: Analysis of student projects in a software engineering course. 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET).



## ДОДАТОК А

### Планування робіт

#### Деталізація мети методом SMART.

S – конкретність, специфічність – розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами.

M – вимірюваність – розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами за 35 днів.

A – узгодженість – розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами на основі існуючих інструментів і технологій.

R – реалістичність, релевантність – розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами урахувавши бюджет, часові та матеріальні ресурси, ризики.

T – обмеженість в часі – розробити інформаційну технологію підтримки діяльності оператора пульта спостереження за складними об'єктами на основі сформованого календарного плану проекту.

#### Планування змісту структури робіт IT-проекту.

WBS (Work Breakdown Structure) - це інструмент управління проектами, що допомагає розкласти проект на менші, керовані етапи. Він представляє собою ієрархічну декомпозицію проекту на менші, керовані елементи, які називаються робочими пакетами.

Кожен робочий пакет описує роботу, яку потрібно виконати, для досягнення конкретної мети проекту. Робочі пакети можуть бути розподілені між різними командами проекту або виконувачами, що дозволяє керівництву проекту ефективніше контролювати процес виконання проекту та планувати роботу на кожному етапі.

WBS також дозволяє зрозуміти, які завдання потрібно виконати для досягнення мети проекту, та забезпечує можливість оцінювати час і зусилля, необхідні для виконання кожного робочого пакета. В цілому, WBS допомагає управляти ризиками проекту, розподіляти ресурси та планувати виконання завдань відповідно до графіка проекту. WBS діаграма проекту зображена на рисунку А.1.

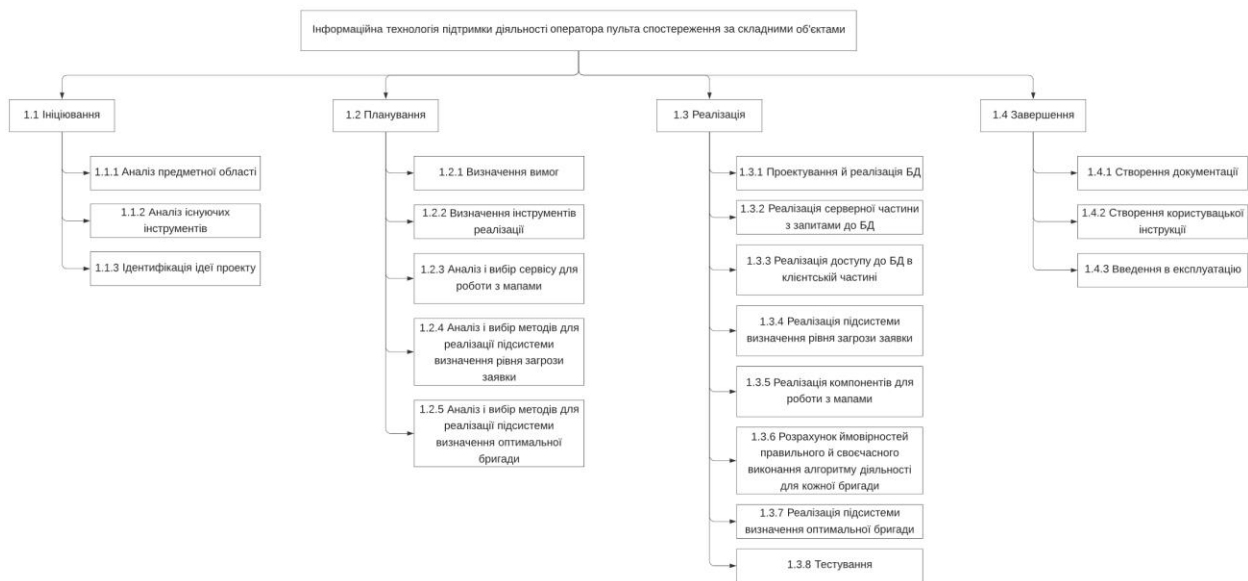


Рисунок А.1 – WBS діаграма проекту

### Планування структури організації, для впровадження готового проекту.

OBS (Organizational Breakdown Structure) - це інструмент управління проектами, який відображає ієрархію структури організації, в якій виконується проект.

Організаційна структура може бути досить складною, з багатьма рівнями управління, департаментами та командами. OBS допомагає упорядкувати цю складну структуру, розкладаючи її на окремі рівні і групуючи відповідні функціональні області, що мають схожі цілі та виконують подібні завдання.

Це дозволяє керівництву проекту легше розуміти, як організація функціонує та хто є відповідальним за кожну частину проекту. Також це

допомагає забезпечити ефективну комунікацію та співпрацю між різними департаментами та командами, що забезпечує успішне виконання проекту.

Крім того, OBS може бути використана для розподілу витрат та контролю витрат на кожній рівні організації. Це дозволяє керівництву проекту більш точно оцінити витрати та визначити, де можна зменшити витрати, не зашкоджуючи проекту. OBS-структуру проекту представлено на рисунку А.2.

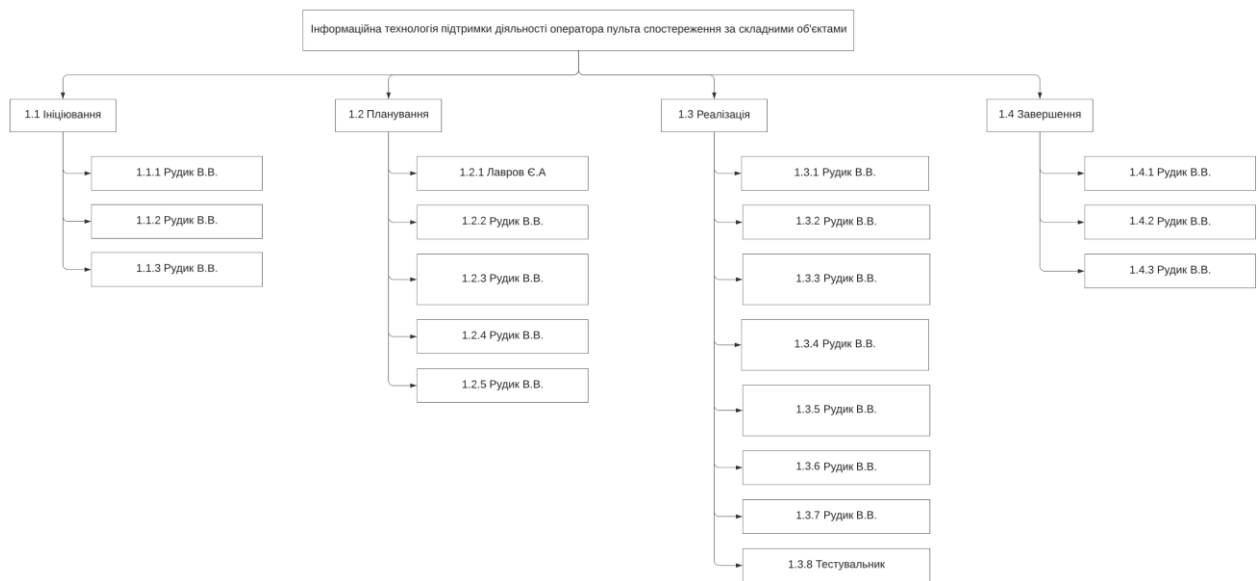


Рисунок А.2 – OBS-структура проекту

### Побудова матриці відповідальності.

Матриця відповідальності - це інструмент управління проектами, який допомагає визначити ролі та відповідальності кожного учасника проекту в рамках конкретних завдань. Вона допомагає забезпечити ясність щодо того, хто відповідає за які завдання та які ресурси необхідні для їх виконання. Це забезпечує ефективну комунікацію та співпрацю між учасниками проекту, допомагає уникнути дублювання робіт, зменшити можливість помилок та забезпечити вчасне виконання завдань.

Для ІТ проектів дуже важливе використання матриці відповідальності, оскільки часто буває багато учасників та складний технічний стек, що потребує чіткого розподілу відповідальності та контролю над роботою.

Таким чином, на підставі OBS та WBS структур побудовано матрицю відповідальності проекту, вона закріплює за кожною елементарною роботою виконавця. Матрицю відповідальності проекту представлено в таблиці А.1.

Таблиця А.1 – Матриця відповідальності

WBS	OBS	Лавров Є.А.	Рудик В.В.	Тестувальник
1.1	1.1.1		X	
	1.1.2		X	
	1.1.3		X	
1.2	1.2.1	X		
	1.2.2		X	
	1.2.3		X	
	1.2.4		X	
	1.2.5		X	
1.3	1.3.1		X	
	1.3.2		X	
	1.3.3		X	
	1.3.4		X	
	1.3.5		X	
	1.3.6		X	
	1.3.7		X	
	1.3.8			X
1.4	1.4.1		X	
	1.4.2		X	
	1.4.3		X	

### **Побудова календарного графіку й діаграми Ганта проекту.**

Календарний графік створення ІТ проекту - це план проекту, що відображає послідовність подій, завдань та термінів, необхідних для успішного виконання проекту. Він може містити такі елементи, як терміни виконання завдань, мілістоуни, майлстоуни, залежності між завданнями, критичний шлях і т.д.

Календарний графік є важливим інструментом управління проектами, який допомагає планувати, відстежувати та контролювати прогрес проекту. Він дозволяє керівництву проекту і команді проекту зрозуміти, коли повинні бути виконані різні етапи проекту, та забезпечити вчасне виконання завдань. Крім того, календарний графік допомагає визначити залежності між завданнями та ідентифікувати критичні шляхи проекту, що може допомогти зменшити ризики та покращити керування проектом.

У контексті ІТ проектів, календарний графік допомагає визначити терміни виконання програмних модулів, тестування, релізів та інших етапів проекту. Крім того, він може допомогти вирішити проблеми зі звуженням термінів та залежностями між різними етапами проекту, що може допомогти зберегти ресурси та виконати проект вчасно та ефективно. Результат побудови календарного плану проекту зображений на рисунку А.3.

i	Task Mode	Назва задачі	Duration	Resource Names	Start	Finish	Predecessors	Cost
1		Інформаційна технологія підтримки діяльності оператора пульта спостереження за складними об'єктами	35 days		Mon 20.03.23	Fri 05.05.23		9 000,00 €
2		Ініціювання	3 days		Mon 20.03.23	Wed 22.03.23		720,00 €
3		Аналіз предметної області	1 day	Рудик В.В.	Mon 20.03.23	Mon 20.03.23		240,00 €
4		Аналіз існуючих інструментів	1 day	Рудик В.В.	Tue 21.03.23	Tue 21.03.23	3	240,00 €
5		Ідентифікація ідей проекту	1 day	Рудик В.В.	Wed 22.03.23	Wed 22.03.23	4	240,00 €
6		Планування	6 days		Thu 23.03.23	Thu 30.03.23	2	2 160,00 €
7		Визначення вимог	2 days	Лавров Є.А.	Thu 23.03.23	Fri 24.03.23		1 200,00 €
8		Визначення інструментів реалізації	1 day	Рудик В.В.	Mon 27.03.23	Mon 27.03.23	7	240,00 €
9		Аналіз і вибір сервісу для роботи з мапа	1 day	Рудик В.В.	Tue 28.03.23	Tue 28.03.23	8	240,00 €
10		Аналіз і вибір методів для реалізації під	1 day	Рудик В.В.	Wed 29.03.23	Wed 29.03.23	9	240,00 €
11		Аналіз і вибір методів для реалізації під	1 day	Рудик В.В.	Thu 30.03.23	Thu 30.03.23	10	240,00 €
12		Реалізація	23 days		Fri 31.03.23	Tue 02.05.23	6	5 400,00 €
13		Проектування й реалізація БД	2 days	Рудик В.В.	Fri 31.03.23	Mon 03.04.23		480,00 €
14		Реалізація серверної частини з запитами до БД	4 days	Рудик В.В.	Tue 04.04.23	Fri 07.04.23	13	960,00 €
15		Реалізація доступу до БД в клієнтській частині	3 days	Рудик В.В.	Mon 10.04.23	Wed 12.04.23	14	720,00 €
16		Реалізація підсистеми визначення рівня загрози заявки	3 days	Рудик В.В.	Thu 13.04.23	Mon 17.04.23	15	720,00 €
17		Реалізація компонентів для роботи з мапами	2 days	Рудик В.В.	Tue 18.04.23	Wed 19.04.23	16	480,00 €
18		Розрахунок ймовірностей правильного й своєчасного виконання алгоритму	2 days	Рудик В.В.	Thu 20.04.23	Fri 21.04.23	17	480,00 €
19		Реалізація підсистеми визначення опти	5 days	Рудик В.В.	Mon 24.04.23	Fri 28.04.23	18	1 200,00 €
20		Тестування	2 days	Тестувальник	Mon 01.05.23	Tue 02.05.23	19	360,00 €
21		Завершення	3 days		Wed 03.05.23	Fri 05.05.23	12	720,00 €
22		Створення документації	1 day	Рудик В.В.	Wed 03.05.23	Wed 03.05.23		240,00 €
23		Створення користувацької інструкції	1 day	Рудик В.В.	Thu 04.05.23	Thu 04.05.23	22	240,00 €
24		Введення в експлуатацію	1 day	Рудик В.В.	Fri 05.05.23	Fri 05.05.23	23	240,00 €

Рисунок А.3 – Календарний графік проекту

Діаграма Ганта - це інструмент в проектному менеджменті, який візуально відображає розклад проекту у вигляді стовпців, що представляють часовий проміжок, та горизонтальних ліній, що представляють різні завдання чи етапи

проекту. Вона допомагає проектному менеджеру і учасникам проекту відслідковувати виконання завдань, керувати ресурсами та прогресом проекту.

Діаграма Ганта відображає графічно календарний графік проекту з усіма завданнями та їхнім тривалістю, а також датами початку та закінчення. Завдяки цьому, всі учасники проекту можуть відстежувати, що потрібно робити, коли це повинно бути зроблено та хто відповідальний за кожне завдання. Крім того, проектний менеджер може відстежувати виконання завдань та їхній прогрес у часі, що дозволяє швидко виявляти будь-які проблеми або затримки та приймати необхідні заходи для вирішення цих проблем. Діаграма також може бути корисна для візуалізації критичного шляху проекту - це найбільш важливі етапи або завдання, які мають великий вплив на загальний прогрес проекту. Завдяки цьому, проектний менеджер може виявити найважливіші завдання та зосередитися на їхньому успішному виконанні.

Узагальнюючи, діаграма Ганта дозволяє проектному менеджеру та учасникам проекту бачити повний зображення проекту, контролювати виконання завдань, керувати ресурсами та вирішувати будь-які проблеми, що можуть виникнути в процесі роботи над проектом. Діаграма Ганта проекту зображена на рисунку А.4.

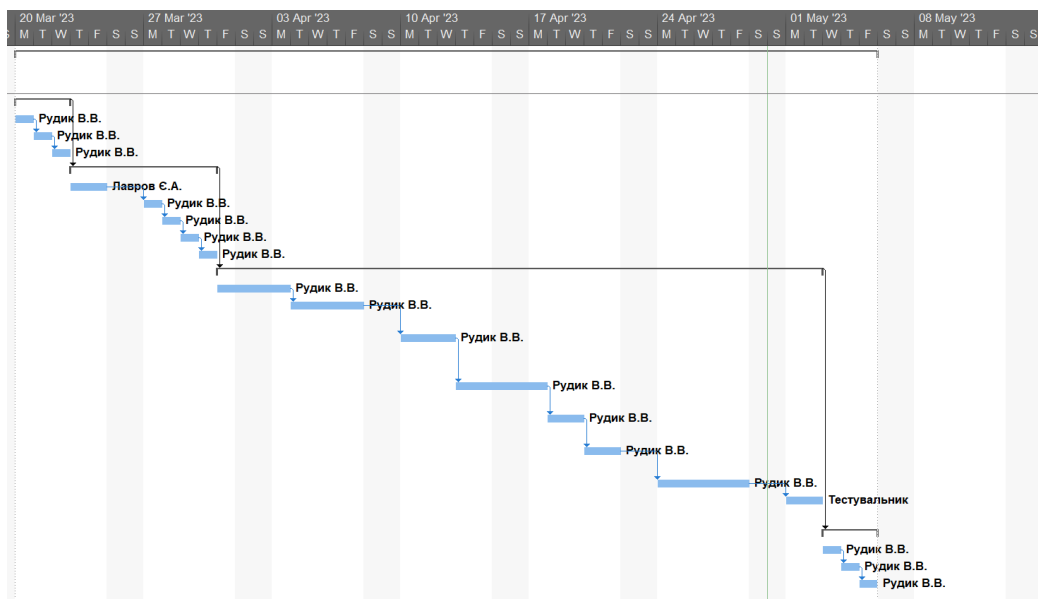
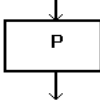
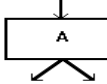
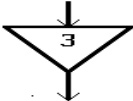
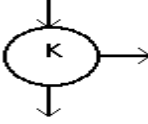
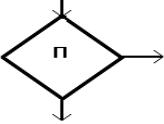
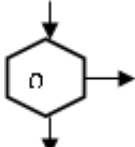


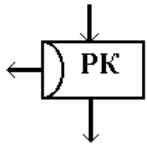
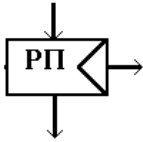
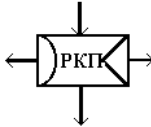
Рисунок А.4 – Діаграма Ганта

## ДОДАТОК Б

Таблиця Б.1- Типові функціональні одиниці.

Види функціонерів	Назва	Умовне позначення
Основні	Робочий	
	Логічний (Альтернативний)	
	Затримки	
Допоміжні	Функціональний контроль	
	Діагностичний контроль	
	Організаційний контроль	

Продовження таблиці Б.1.

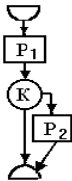
Назва	Умовне позначення
Робочий з одночасним контролем результатів функціонування	
Робочий з одночасною діагностикою техніки	
Робочий з одночасним контролем результатів функціонування і діагностикою техніки	

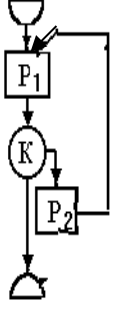


## ДОДАТОК В

Таблиця В.1 - Типові функціональні структури.

Зміст типової функціональної структури (ТФС)	Схема ТФС	Показник	Розрахункова формула
1. Послідовне виконання робочих операцій, RR		Ймовірність безпомилкового виконання операції	$B = \prod_{i=1}^n B_i$
		Математичне очікування часу виконання операції (трудомісткості, вартості)	$M(X) = \sum_{i=1}^n M(X_i)$ $X = \{T, W, C\}$
		Дисперсія часу виконання операції (трудомісткості, вартості)	$D(X) = \sum_{i=1}^n D(X_i)$ $X = \{T, W, C\}$
2. Циклічна функціональна схема "Робоча операція з контролем функціонування без обмеження на кількість циклів", RK		Ймовірність безпомилкового виконання операції	$B = B^1 * K^{11} * \frac{1}{1 - (B^1 * K^{10} + B^0 * K^{00})}$
Математичне очікування часу виконання операції (трудомісткості, вартості)	$M(X) = (M(X_p) + M(X_k)) * M(L)$ $M(L) = \frac{1}{1 - (B^1 * K^{10} + B^0 * K^{00})}$ $X = \{T, W, C\}$		

Зміст типової функціональної структури (ТФС)	Схема ТФС	Показник	Розрахункова формула
		Дисперсія часу виконання операції (трудомісткості, вартості)	$D(X) = D(L) * (M(X_p) + M(X_k))^2 + (D(X_p) + D(X_k)) * M(L)$ $D(L) = \frac{B^1 * K^{10} + B^0 * K^{00}}{(1 - (B^1 * K^{10} + B^0 * K^{00}))^2}$ $X = \{T, W, C\}$
Робоча операція з контролем функціонування і виправленням помилки без циклів, <i>RKRI</i>		Ймовірність безпомилкового виконання операції	$B = \frac{B_1^1 * K^{11}}{(B_1^0 * K^{00} + B_1^1 * K^{10}) * B_2^1}$
		Математичне очікування часу виконання операції (трудомісткості, вартості)	$M(X) = M(X_{p1}) + M(X_k) + (B_1^0 * K^{00} + B_1^1 * K^{10}) * M(X_{p2})$ $X = \{T, W, C\}$
		Дисперсія часу виконання операції (трудомісткості, вартості)	$D(X) = D(X_{p1}) + D(X_k) + (B_1^0 * K^{00} + B_1^1 * K^{10}) * D(X_{p2}) + (B_1^0 * K^{00} + B_1^1 * K^{10}) * (B_1^1 * K^{11} + B_1^0 * K^{01}) * M^2(X_{p2})$ $X = \{T, W, C\}$
Циклічна ФС «Робоча операція з контролем		Ймовірність безпомилкового виконання операції	$B = \frac{B_1^1 * K^{11} (1 - K^{00} * B_2^0)}{K^{01} + B_1^1 * B_2^1 (K^{11} - K^{01})}$

Зміст типової функціональної структури (ТФС)	Схема ТФС	Показник	Розрахункова формула
функціонування, виправленням та повторенням робочої операції без обмеження на кількість циклів», <i>RKR</i>		Математичне очікування часу виконання операції (трудомісткості, вартості)	$M(X) = M(X_{p1}) + M(X_k) + [M(X_{p1}) + M(X_{p2}) + M(X_k)]^*$ $* \frac{B^1 * K^{10} + B^0 * K^{00}}{1 - (B^1 K^{10} + B^0 * K^{00})}, \text{ де}$ $B^1 = B_1^1 * B_2^1; \quad B^0 = 1 - B^1$
		Дисперсія часу виконання операції (трудомісткості, вартості)	$D(T) = D(X_{p1}) + D(X_k) + [D(X_{p1}) + D(X_{p2}) + D(X_k)]^*$ $* \frac{B^1 * K^{10} + B_1^0 * K^{00}}{1 - (B^1 K^{10} + B^0 * K^{00})} +$ $+ \frac{B_1^1 * K^{10} + B_1^0 * K^{00}}{(1 - (B^1 K^{10} + B^0 * K^{00}))^2} *$ $* \frac{B_1^1 * K^{10} + B_1^0 * K^{00}}{1 - (B^1 K^{10} + B^0 * K^{00})} *$ $* \frac{B_1^1 * K^{11} + B_1^0 * K^{01}}{(1 - (B^1 K^{10} + B^0 * K^{00}))^2} *$ $* [M(X_{p1}) + M(X_{p2}) + M(X_k)]^2$ <p>где <math>X = \{T, W, C\}</math></p>

## ДОДАТОК Г

Фрагменти коду реалізації інформаційної технології.

**index.js (backend):**

```
import express from 'express'
import mysql from 'mysql'
import cors from "cors";

const app = express()
app.use(cors());
app.use(express.json());

const db = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "operator_helper",
});

//test
app.get("/", (request, response) => {
  response.json("test");
});

//get objects
app.get("/objects", (request, response) => {
  const query = "SELECT * FROM objects"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//get object by id
app.get("/objects/:id", (request, response) => {
  const objectId = request.params.id;
  const query = "SELECT * FROM objects WHERE id = ? "
  db.query(query, [objectId], (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});
```

```

    })
  });

//add object
app.post("/objects", (request, response) => {
  const query = "INSERT INTO objects(`id`, `street`, `house`,
`section`, `floor`, `apartment`, `latitude`, `longitude`,
`category`, `pets`, `client_id`, `object_is_active`) VALUES (?)"

  const values = [
    request.body.id,
    request.body.street,
    request.body.house,
    request.body.section,
    request.body.floor,
    request.body.apartment,
    request.body.latitude,
    request.body.longitude,
    request.body.category,
    request.body.pets,
    request.body.client_id,
    request.body.object_is_active,
  ];

  db.query(query, [values], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//delete object
app.delete("/objects/:id", (request, response) => {
  const objectId = request.params.id;
  const query = " DELETE FROM objects WHERE id = ? ";

  db.query(query, [objectId], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//update object
app.put("/objects/:id", (request, response) => {
  const objectId = request.params.id;
  const query = "UPDATE objects SET `id` = ?, `street` = ?,
`house` = ?, `section` = ?, `floor` = ?, `apartment` = ?,
`latitude` = ?, `longitude` = ?, `category` = ?, `pets` = ?,
`client_id` = ?, `object_is_active` = ? WHERE `id` = ?";

  const values = [
    request.body.street,
    request.body.house,

```

```

    request.body.section,
    request.body.floor,
    request.body.apartment,
    request.body.latitude,
    request.body.longitude,
    request.body.category,
    request.body.pets,
    request.body.client_id,
    request.body.object_is_active,
  ];

  db.query(query, [objectId, ...values, objectId], (error, data)
=> {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//add client
app.post("/clients", (request, response) => {
  const query = "INSERT INTO clients(`id`, `surname`, `name`,
`address`, `phone`, `alternate_phone`) VALUES (?)";

  const values = [
    request.body.id,
    request.body.surname,
    request.body.name,
    request.body.address,
    request.body.phone,
    request.body.alternate_phone,
  ];

  db.query(query, [values], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//update client
app.put("/clients/:id", (request, response) => {
  const clientId = request.params.id;
  const query = "UPDATE clients SET `id` = ?, `surname` = ?,
`name` = ?, `address` = ?, `phone` = ?, `alternate_phone` = ?
WHERE `id` = ?";

  const values = [
    request.body.surname,
    request.body.name,
    request.body.address,
    request.body.phone,
    request.body.alternate_phone,
  ];

```

```

];

db.query(query, [clientId, ...values, clientId], (error, data)
=> {
  if (error) return response.send(error);
  return response.json(data);
});
});

//get clients
app.get("/clients", (request, response) => {
  const query = "SELECT * FROM clients"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//get client by id
app.get("/clients/:id", (request, response) => {
  const clientId = request.params.id;
  const query = "SELECT * FROM clients WHERE id = ? "
  db.query(query, [clientId], (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//delete object
app.delete("/clients/:id", (request, response) => {
  const clientId = request.params.id;
  const query = " DELETE FROM clients WHERE id = ? ";

  db.query(query, [clientId], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//get patrolmen
app.get("/patrolmen", (request, response) => {
  const query = "SELECT id, full_name, DATE_FORMAT(age, '%d/%m/%Y')
AS age, DATE_FORMAT(experience, '%d/%m/%Y') AS experience FROM
patrolmen"
  db.query(query, (error, data) => {
    if (error) {

```

```

        console.log(error);
        return response.json(error);
    }
    return response.json(data);
  })
});

//get patrolman by id
app.get("/patrolmen/:id", (request, response) => {
  const patrolmenId = request.params.id;
  const query = "SELECT id, full_name, DATE_FORMAT(age, '%Y-%m-%d')
AS age, DATE_FORMAT(experience, '%Y-%m-%d') AS experience FROM
patrolmen WHERE id = ? "
  db.query(query, [patrolmenId], (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//add patrolman
app.post("/patrolmen", (request, response) => {
  const query = "INSERT INTO patrolmen(`id`, `full_name`, `age`,
`experience`) VALUES (?)";

  const values = [
    request.body.id,
    request.body.full_name,
    request.body.age,
    request.body.experience,
  ];

  db.query(query, [values], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//delete patrolmen
app.delete("/patrolmen/:id", (request, response) => {
  const patrolmenId = request.params.id;
  const query = " DELETE FROM patrolmen WHERE id = ? ";

  db.query(query, [patrolmenId], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//update patrolmen

```



```

app.put("/patrolmen/:id", (request, response) => {
  const patrolmenId = request.params.id;
  const query = "UPDATE patrolmen SET `id` = ?, `full_name` = ?,
`age` = ?, `experience` = ? WHERE `id` = ?";

  const values = [
    request.body.full_name,
    request.body.age,
    request.body.experience,
  ];

  db.query(query, [patrolmenId, ...values, patrolmenId], (error,
data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//get patrols
app.get("/patrols", (request, response) => {
  const query = "SELECT * FROM patrols"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//get patrols with patrolmen names
app.get("/patrols-name", (request, response) => {
  const query = "SELECT patrols.id, patrols.probability_superlow
as patrolprobabilitysuperlow, patrols.probability_low as
patrolprobabilitylow, patrols.probability_medium as
patrolprobabilitymedium, patrols.probability_high as
patrolprobabilityhigh, patrols.first_patrolman_id as
firstpatrolmanid, patrols.second_patrolman_id as
secondpatrolmanid, patrols.latitude as latitude, patrols.longitude
as longitude, (SELECT patrolmen.full_name from patrolmen where
patrolmen.id = patrols.first_patrolman_id) as firstpatrolmanname,
(SELECT patrolmen.full_name from patrolmen where patrolmen.id =
patrols.second_patrolman_id) as secondpatrolmanname FROM patrols
where patrol_is_active = 1"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

```

```

//get patrol by id
app.get("/patrols/:id", (request, response) => {
  const patrolId = request.params.id;
  const query = "SELECT * FROM patrols WHERE id = ? "
  db.query(query, [patrolId], (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//add patrol
app.post("/patrols", (request, response) => {
  const query = "INSERT INTO patrols(`id`, `first_patrolman_id`,
`second_patrolman_id`, `latitude`, `longitude`,
`patrol_is_active`, `probability_superlow`, `probability_low`,
`probability_medium`, `probability_high`) VALUES (?)";

  const values = [
    request.body.id,
    request.body.first_patrolman_id,
    request.body.second_patrolman_id,
    request.body.latitude,
    request.body.longitude,
    request.body.patrol_is_active,
    request.body.probabilitysuperlow,
    request.body.probabilitylow,
    request.body.probabilitymedium,
    request.body.probabilityhigh,
  ];

  db.query(query, [values], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//delete patrol
app.delete("/patrols/:id", (request, response) => {
  const patrolId = request.params.id;
  const query = " DELETE FROM patrols WHERE id = ? ";

  db.query(query, [patrolId], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//update patrol

```

```

app.put("/patrols/:id", (request, response) => {
  const patrolId = request.params.id;
  const query = "UPDATE patrols SET `id` = ?, `first_patrolman_id`
= ?, `second_patrolman_id` = ?, `latitude` = ?, `longitude` = ?,
`patrol_is_active` = ?, `probability_superlow` = ? ,
`probability_low` = ?, `probability_medium` = ?,
`probability_high` = ? WHERE `id` = ?";

  const values = [
    request.body.first_patrolman_id,
    request.body.second_patrolman_id,
    request.body.latitude,
    request.body.longitude,
    request.body.patrol_is_active,
    request.body.probability_superlow,
    request.body.probability_low,
    request.body.probability_medium,
    request.body.probability_high,
  ];

  db.query(query, [patrolId, ...values, patrolId], (error, data)
=> {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//get incidents
app.get("/incidents", (request, response) => {
  const query = "SELECT * FROM incidents"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//get incidents with join
app.get("/incidents-join", (request, response) => {
  const query = "SELECT incidents.id AS incidentid,
incidents.zones AS incidentzones, incidents.danger AS
incidentdanger, incidents.is_checked AS incidentchecked,
incidents.iterations AS incidentiterations, clients.id AS
clientid, CONCAT(clients.surname, ' ', clients.name) AS
clientname, clients.address AS clientaddress, clients.phone AS
clientphone, clients.alternate_phone AS clientalternatephone,
objects.id as objectid, objects.latitude AS objectlatitude,
objects.longitude AS objectlongitude, objects.street AS
objectstreet, objects.house AS objecthouse, objects.section AS
objectsection, objects.floor AS objectfloor, objects.apartment AS

```

```

objectapartment, objects.category AS objectcategory, objects.pets
as objectpets, timestamps.id AS timestampid,
DATE_FORMAT(timestamps.date, '%d.%m.%Y') AS date, timestamps.time
AS time FROM incidents, clients, timestamps, objects WHERE
incidents.timestamp_id=timestamps.id and
incidents.object_id=objects.id and objects.client_id=clients.id"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

```

```

//get opened incidents with join
app.get("/opened-incidents", (request, response) => {
  const query = "SELECT incidents.id AS incidentid,
incidents.zones AS incidentzones, incidents.danger AS
incidentdanger, incidents.is_checked AS incidentchecked,
incidents.iterations AS incidentiterations, clients.id AS
clientid, CONCAT(clients.surname, ' ', clients.name) AS
clientname, clients.address AS clientaddress, clients.phone AS
clientphone, clients.alternate_phone AS clientalternatephone,
objects.id as objectid, objects.latitude AS objectlatitude,
objects.longitude AS objectlongitude, objects.street AS
objectstreet, objects.house AS objecthouse, objects.section AS
objectsection, objects.floor AS objectfloor, objects.apartment AS
objectapartment, objects.category AS objectcategory, objects.pets
as objectpets, timestamps.id AS timestampid,
DATE_FORMAT(timestamps.date, '%d.%m.%Y') AS date, timestamps.time
AS time FROM incidents, clients, timestamps, objects WHERE
incidents.timestamp_id=timestamps.id and
incidents.object_id=objects.id and objects.client_id=clients.id
and incidents.is_closed=0"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

```

```

//get unchecked incidents
app.get("/unchecked-incidents", (request, response) => {
  const query = "SELECT * FROM incidents where `is_checked` = '0'"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
  })
});

```

```

    return response.json(data);
  })
});

//set incident checked
app.put("/check-incident/:id", (request, response) => {
  const incidentId = request.params.id;
  const query = "UPDATE incidents SET `is_checked` = 1 WHERE `id`
= ?";

  db.query(query, [incidentId], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//get incident by id
app.get("/incidents/:id", (request, response) => {
  const incidentId = request.params.id;
  const query = "SELECT incidents.id AS incidentid,
incidents.zones AS incidentzones, incidents.danger AS
incidentdanger, incidents.is_checked AS incidentchecked,
incidents.iterations AS incidentiterations, clients.id AS
clientid, CONCAT(clients.surname, ' ', clients.name) AS
clientname, clients.address AS clientaddress, clients.phone AS
clientphone, clients.alternate_phone AS clientalternatephone,
objects.id as objectid, objects.latitude AS objectlatitude,
objects.longitude AS objectlongitude, objects.street AS
objectstreet, objects.house AS objecthouse, objects.section AS
objectsection, objects.floor AS objectfloor, objects.apartment AS
objectapartment, objects.category AS objectcategory, objects.pets
as objectpets, timestamps.id AS timestampid,
DATE_FORMAT(timestamps.date, '%d.%m.%Y') AS date, timestamps.time
AS time FROM incidents, clients, timestamps, objects WHERE
incidents.timestamp_id=timestamps.id and
incidents.object_id=objects.id and objects.client_id=clients.id
and incidents.id = ? "
  db.query(query, [incidentId], (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//set danger for incident
app.put("/incident-danger/:id", (request, response) => {
  const incidentId = request.params.id;
  const query = "UPDATE incidents SET `danger` = ? WHERE `id` =
?";

```

```

const values = [
  request.body.danger,
];

db.query(query, [...values, incidentId], (error, data) => {
  if (error) return response.send(error);
  return response.json(data);
});
});

//set is_checked switch = 1 for incident
app.put("/incident-ischeck/:id", (request, response) => {
  const incidentId = request.params.id;
  const query = "UPDATE incidents SET `is_checked` = 1 WHERE `id`
= ?";

  db.query(query, [incidentId], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });
});

//get client by id
app.get("/clients/:id", (request, response) => {
  const clientId = request.params.id;
  const query = "SELECT * FROM clients WHERE id = ? "
  db.query(query, [clientId], (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//get last operation of the incident by id
app.get("/last-operation/:id", (request, response) => {
  const incidentId = request.params.id;
  const query = "SELECT operations.id as operationid,
operations.incident_id as incidentid, operations.patrol_id as
patrolid, operations.status_id as statusid,
DATE_FORMAT(timestamps.date, '%d.%m.%Y') AS date, timestamps.time
AS time, statuses.title as statustitle, statuses.description as
statusdescription, patrols.first_patrolman_id as firstpatrolmanid,
patrols.second_patrolman_id as secondpatrolmanid, patrols.latitude
as patrollatitude, patrols.longitude as patrollongitude, (SELECT
patrolmen.full_name from patrolmen where
id=patrols.first_patrolman_id) as firstpatrolmanname, (SELECT
patrolmen.full_name from patrolmen where
id=patrols.second_patrolman_id) as secondpatrolmanname, (SELECT
DATE_FORMAT(patrolmen.age, '%d/%m/%Y') from patrolmen where
id=patrols.first_patrolman_id) as firstpatrolmanage, (SELECT

```

```

DATE_FORMAT(patrolmen.age,'%d/%m/%Y') from patrolmen where
id=patrols.second_patrolman_id) as secondpatrolmanage, (SELECT
DATE_FORMAT(patrolmen.experience,'%d/%m/%Y') from patrolmen where
id=patrols.first_patrolman_id) as firstpatrolmanexp, (SELECT
DATE_FORMAT(patrolmen.experience,'%d/%m/%Y') from patrolmen where
id=patrols.second_patrolman_id) as secondpatrolmanexp FROM
operations, timestamps, statuses, patrols, patrolmen WHERE
operations.incident_id = ? and
operations.timestamp_id=timestamps.id and operations.status_id =
statuses.id and operations.patrol_id = patrols.id ORDER BY
operations.id DESC LIMIT 1"
  db.query(query, [incidentId], (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//get statuses
app.get("/statuses", (request, response) => {
  const query = "SELECT * FROM statuses"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//add operation
app.post("/operations", (request, response) => {
  const query = "INSERT INTO operations(`id`, `incident_id`,
`patrol_id`, `status_id`, `timestamp_id`) VALUES (?";

  const values = [
    request.body.id,
    request.body.incident_id,
    request.body.patrol_id,
    request.body.status_id,
    request.body.timestamp_id,
  ];

  db.query(query, [values], (error, data) => {
    if (error) return response.send(error);
    return response.json(data);
  });

  if (request.body.status_id === 50) {

```

```

    db.query("UPDATE incidents SET `is_closed` = 1 WHERE `id` =
?", request.body.incident_id);
    //db.query("UPDATE patrols SET `patrol_is_active` = 1 WHERE
`id` = ?", request.body.patrol_id);
    } else {
    db.query("UPDATE incidents SET `is_closed` = 0 WHERE `id` =
?", request.body.incident_id);
    //db.query("UPDATE patrols SET `patrol_is_active` = 0 WHERE
`id` = ?", request.body.patrol_id);
    }
});

//get operations
app.get("/operations", (request, response) => {
    const query = "SELECT * FROM operations"
    db.query(query, (error, data) => {
        if (error) {
            console.log(error);
            return response.json(error);
        }
        return response.json(data);
    })
});

//get last operation id
app.get("/operations-lastid", (request, response) => {
    const query = "SELECT MAX(id) as id FROM operations"
    db.query(query, (error, data) => {
        if (error) {
            console.log(error);
            return response.json(error);
        }
        return response.json(data);
    })
});

//add timestamp
app.post("/timestamps", (request, response) => {
    const query = "INSERT INTO timestamps(`id`) VALUES (?)";

    const values = [
        request.body.id,
    ];

    db.query(query, [values], (error, data) => {
        if (error) return response.send(error);
        return response.json(data);
    });
});

```



```

//get timestamps
app.get("/timestamps", (request, response) => {
  const query = "SELECT * FROM timestamps"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

//get last timestamp id value
app.get("/timestamps-lastid", (request, response) => {
  const query = "SELECT MAX(id) as id FROM timestamps"
  db.query(query, (error, data) => {
    if (error) {
      console.log(error);
      return response.json(error);
    }
    return response.json(data);
  })
});

app.listen(8800, () => {
  console.log("Connected to backend")
})

```

### appRoutes.tsx:

```

import DashboardPageLayout from
"../pages/dashboard/DashboardPageLayout";
import { RouteType } from "../config";
import MapPage from "../pages/dashboard/MapPage";
import DashboardIndex from "../pages/dashboard/DashboardIndex";
import Ahp from "../pages/ahp/Ahp";
import IncidentsPage from "../pages/dashboard/IncidentsPage";
import SaasPage from "../pages/dashboard/SaasPage";
import ManagePageLayout from
"../pages/component/ManagePageLayout";
import DashboardOutlinedIcon from '@mui/icons-
material/DashboardOutlined';
import AppsOutlinedIcon from '@mui/icons-material/AppsOutlined';
import FormatListBulletedOutlinedIcon from '@mui/icons-
material/FormatListBulletedOutlined';
import ObjectsPage from "../pages/component/ObjectsPage";
import PatrolsPage from "../pages/component/PatrolsPage";
import PatrolmenPage from "../pages/component/PatrolmenPage";
import ClientsPage from "../pages/component/ClientsPage";
import IncidentsItemPage from
"../pages/dashboard/IncidentsItemPage";

```

```

const appRoutes: RouteType[] = [
  {
    index: true,
    element: <MapPage />,
    state: "dashboard.map"
  },
  {
    path: "/dashboard",
    element: <DashboardPageLayout />,
    state: "dashboard",
    sidebarProps: {
      displayText: "Компоненти",
      icon: <DashboardOutlinedIcon />
    },
    child: [
      {
        index: true,
        element: <DashboardIndex />,
        state: "dashboard.index"
      },
      {
        path: "/dashboard/map",
        element: <MapPage />,
        state: "dashboard.map",
        sidebarProps: {
          displayText: "Мапа"
        }
      },
      {
        path: "/dashboard/incidents",
        element: <IncidentsPage />,
        state: "dashboard.incidents",
        sidebarProps: {
          displayText: "Інциденти"
        }
      },
      {
        path: "/dashboard/incidents/:id",
        element: <IncidentsItemPage />,
        state: "dashboard.incidentsid",
      },
      {
        path: "/dashboard/saas",
        element: <SaasPage />,
        state: "dashboard.saas",
      }
    ]
  },
  {
    path: "/manage",

```

```

element: <ManagePageLayout />,
state: "manage",
sidebarProps: {
  displayText: "Управління",
  icon: <AppsOutlinedIcon />
},
child: [
  {
    path: "/manage/objects",
    element: <ObjectsPage />,
    state: "manage.objects",
    sidebarProps: {
      displayText: "Об'єкти"
    },
  },
  {
    path: "/manage/patrols",
    element: <PatrolsPage />,
    state: "manage.patrols",
    sidebarProps: {
      displayText: "Бригади"
    }
  },
  {
    path: "/manage/patrolmen",
    element: <PatrolmenPage />,
    state: "manage.patrolmen",
    sidebarProps: {
      displayText: "Патрульні"
    }
  },
  {
    path: "/manage/clients",
    element: <ClientsPage />,
    state: "manage.clients",
    sidebarProps: {
      displayText: "Клієнти"
    }
  }
]
},
{
  path: "/ahp",
  element: <Ahp />,
  state: "ahp",
  sidebarProps: {
    displayText: "MAI",
    icon: <FormatListBulletedOutlinedIcon />
  }
}
];

```

```
export default appRoutes;
```

### MapPage.jsx:

```
import { useRef, useState, useContext, useEffect } from 'react';
import { MapContainer, TileLayer, Popup, Marker } from 'react-leaflet'
import '../styles/App.css'
import PatrolMarker from "../components/formap/PatrolMarker";
import OptimalRoute from "../components/formap/OptimalRoute";
import Patrols from "../storage/Patrols.json";
import Routes from "../storage/Routes.json";
import { ObjectsContext } from '../context';
import axios from "axios";
import { latLng } from 'leaflet';
import { Toaster } from 'react-hot-toast';

const MapPage = () => {

  const VISICOM_API_KEY = process.env.REACT_APP_VISICOM_API_KEY;
  const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;

  const [center, setCenter] = useState(['50.90068', '34.81324']);

  const mapRef = useRef();

  const mapZoom = 13;
  const scrollWheelZoom = true;

  const [objects, setObjects] = useState([])
  const [patrols, setPatrols] = useState([])

  useEffect(() => {
    const fetchAllObjects = async () => {
      try {
        const response = await axios.get(BECKEND_URL+"/objects");
        setObjects(response.data);
      } catch (error) {
        console.log(error);
      }
    };
    const fetchAllPatrols = async () => {
      try {
        const response = await axios.get(BECKEND_URL+"/patrols");
        setPatrols(response.data);
      } catch (error) {
        console.log(error);
      }
    };
    fetchAllObjects();
    fetchAllPatrols();
  });
}
```

```

}, [BECKEND_URL]);

const [checkRoute, setCheckRoutes] = useState(Routes.routes)

return (
  <div className="Map">
    <Toaster ></Toaster>
    <MapContainer dragging={scrollWheelZoom} center={center}
zoom={mapZoom} ref={mapRef}>
      <TileLayer
        attribution='Дані карт © AT «<a
href="https://api.visicom.ua/">Visicom</a>»'
url={"https://tms{s}.visicom.ua/2.0.0/ua/base/{z}/{x}/{y}.png?key=
" + VISICOM_API_KEY}
        subdomains={'123'}
        tms={true}
      />

      {patrols.map(patrol =>
        patrol.patrol_is_active !== 0 && <PatrolMarker
patrol={patrol} key={patrol.id} />
      )}

      {objects.map(object =>
        object.object_is_active !== 0 &&
        <Marker key={object.id}
position={latLng(object.latitude, object.longitude)}>
          <Popup>
            Об'єкт {object.id}
          </Popup>
        </Marker>
      )}

    </MapContainer>

  </div>

);
}

export default MapPage;

```

### IncidentsItemPage.jsx:

```

import React, { useState, useEffect } from 'react'
import { useNavigate } from "react-router-dom";
import { useParams } from 'react-router-dom';
import axios from "axios";

```

```

import { useRef } from 'react';
import { MapContainer, TileLayer, Marker, GeoJSON } from 'react-
leaflet'
import AHP from 'ahp';
import PatrolMarker from '../../components/formap/PatrolMarker';
import toast, { Toaster } from 'react-hot-toast';

const IncidentsItemPage = () => {

  let navigate = useNavigate();

  const errorAhpInput = () => toast.error("Помилка. Введіть від
1 до 9, або дробом в форматі 1/[1-9]");
  const sucessNotify = () => toast.success("Оновлено!");

  const VISICOM_API_KEY = process.env.REACT_APP_VISICOM_API_KEY;
  const coordinates = ['50.90068', '34.81324']
  const mapRef = useRef();
  const mapZoom = 12;
  const scrollWheelZoom = true;

  const params = useParams()

  const [incident, setIncidents] = useState([])
  const [patrols, setPatrols] = useState([])
  const [patrol, setPatrol] = useState([])
  const [operation, setOperaton] = useState([])
  const [routes, setRoutes] = useState([])
  const [statuses, setStatuses] = useState([])
  const [status, setStatus] = useState([])
  const [ahpResult, setAhpResult] = useState([])
  const [timeResult, setTimeResult] = useState([])
  const [probabilityResult, setProbabilityResult] = useState([])
  const ahpResultArray = [];
  const timeResultArray = [];
  const probabilityResultArray = [];
  const [ahpClicked, setAhpClicked] = useState(false);
  const [timeClicked, setTimeClicked] = useState(false);
  const [probabilityClicked, setProbabilityClicked] =
useState(false);
  var newOperation;
  var newTimestamp;

  const ahpContext = new AHP();
  var outputAHP;

  const fetchOperation = async () => {
    try {
      const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;
      const response = await axios.get(BECKEND_URL + "/last-
operation/" + incident.incidentid);
      setOperaton(response.data[0]);
    }
  }
}

```

```

    } catch (error) {
      console.log(error);
    }
  };

  const fetchLastOperation = async () => {
    try {
      const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;
      const response = await axios.get(BECKEND_URL +
"/operations-lastid");
      newOperation = response.data[0].id + 1;
      return response.data;
    } catch (error) {
      console.log(error);
    }
  };

  const fetchLastTimestamp = async () => {
    try {
      const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;
      const response = await axios.get(BECKEND_URL +
"/timestamps-lastid");
      newTimestamp = response.data[0].id + 1;
      return response.data;
    } catch (error) {
      console.log(error);
    }
  };

  const addNewTimestamp = async () => {
    try {
      const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;
      const response = await axios.post(BECKEND_URL +
"/timestamps", { id: newTimestamp });
      return response.data;
    } catch (error) {
      console.log(error);
    }
  };

  const setOperation = async () => {
    try {
      const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;
      var newOperationObject = { id: newOperation,
incident_id: incident.incidentid, patrol_id: patrol.id, status_id:
status.id, timestamp_id: newTimestamp }
      const response = await axios.post(BECKEND_URL +
"/operations", newOperationObject);
      sucessNotify();
      fetchOperation();
      return response.data;
    } catch (error) {

```

```

        console.log(error);
    }

};

const setNewStatus = (e) => {
    e.preventDefault();

    fetchLastOperation()
        .then((data) => fetchLastTimestamp())
        .then((data) => addNewTimestamp())
        .then((data) => setOperation());
}

const calculateAHP = (routes) => {

    var rankCriteriaTimeProbability = prompt("Введіть від 1 до
9 значення критерію 'Час за який бригада приїде' до 'Ймовірність
своєчасного і правильного виконання операції бригадою'")
    var regexp = new RegExp("[1-9]$")
    var isValid = false;

    if (regexp.test(rankCriteriaTimeProbability)) {
        isValid = true;
    } else {
        if ((regexp.test(rankCriteriaTimeProbability[0])) &&
(regexp.test(rankCriteriaTimeProbability[2])) &&
(rankCriteriaTimeProbability[1] === "/" ) &&
(rankCriteriaTimeProbability.length === 3)) {
            isValid = true;
            rankCriteriaTimeProbability =
rankCriteriaTimeProbability[0] / rankCriteriaTimeProbability[2]
        }
    }

    if (isValid) {

        for (var i = 0; i < routes.length; i++) {
            ahpContext.addItem("PatrolID " +
routes[i].patrolid);
        }

        ahpContext.addCriteria(['Time', 'Probability']);

        var arraySize = 0;
        for (i = 1; i < routes.length; i++) {
            arraySize = arraySize + i;
        }
    }
}

```



```

var rankCriteriaTime = new Array(arraySize);
var iterator = 0;
for (i = 0; i < routes.length; i++) {
    for (var j = i + 1; j < routes.length; j++) {
        var time1 = Number((routes[i].distance /
22).toFixed());
        var time2 = Number((routes[j].distance /
22).toFixed());
        rankCriteriaTime[iterator] = ['PatrolID ' +
routes[i].patrolid, 'PatrolID ' + routes[j].patrolid, time2 /
time1];
        iterator++;
    }
}

ahpContext.rankCriteriaItem('Time', rankCriteriaTime);
var rankCriteriaProbability = new Array(arraySize);
iterator = 0;
for (i = 0; i < routes.length; i++) {
    for (j = i + 1; j < routes.length; j++) {
        var prob1 = Number(routes[i].probability);
        var prob2 = Number(routes[j].probability);
        rankCriteriaProbability[iterator] = ['PatrolID
' + routes[i].patrolid, 'PatrolID ' + routes[j].patrolid, prob1 /
prob2];
        iterator++;
    }
}
ahpContext.rankCriteriaItem('Probability',
rankCriteriaProbability);
ahpContext.rankCriteria(
[
    ['Time', 'Probability',
rankCriteriaTimeProbability]
]
);
outputAHP = ahpContext.run();

for (i = 0; i < outputAHP.rankedScores.length; i++) {
    var newScore = { patrolid: routes[i].patrolid,
score: outputAHP.rankedScores[i].toFixed(5), distance:
routes[i].distance, probability: routes[i].probability, time:
Number(routes[i].distance / 22).toFixed() }
    ahpResultArray.push(newScore)
}

ahpResultArray.sort(function (a, b) {
    if (a.score < b.score) {
        return 1;
    }
    if (a.score > b.score) {

```

```

        return -1;
    }
    return 0;
});

setAhpResult(ahpResultArray)

setTimeClicked(false)
setProbabilityClicked(false)

setAhpClicked(true);
} else {
    errorAhpInput();
}
}

const calculateClick = (e) => {
    e.preventDefault();

    calculateAHP(routes)
}

const timeClick = (e) => {
    e.preventDefault();

    calculateByTime(routes)
}

const universalClick = (e) => {
    e.preventDefault();

    let path = `/ahp`;
    navigate(path);
}

const calculateByTime = (routes) => {

    for (var i = 0; i < routes.length; i++) {
        var newScore = { patrolid: routes[i].patrolid,
distance: routes[i].distance, probability: routes[i].probability,
time: Number(routes[i].distance / 22).toFixed() }
        timeResultArray.push(newScore)
    }

    timeResultArray.sort(function (a, b) {
        if (a.distance > b.distance) {
            return 1;
        }
        if (a.distance < b.distance) {
            return -1;
        }
    })
}

```

```

        return 0;
    });

    setTimeResult(timeResultArray)

    setProbabilityClicked(false)
    setAhpClicked(false);
    setTimeClicked(true)
}

const probabilityClick = (e) => {
    e.preventDefault();

    calculateByProbability(routes)
}

const calculateByProbability = (routes) => {

    for (var i = 0; i < routes.length; i++) {
        var newScore = { patrolid: routes[i].patrolid,
distance: routes[i].distance, probability: routes[i].probability,
time: Number(routes[i].distance / 22).toFixed() }
        probabilityResultArray.push(newScore)
    }

    probabilityResultArray.sort(function (a, b) {
        if (a.probability < b.probability) {
            return 1;
        }
        if (a.probability > b.probability) {
            return -1;
        }
        return 0;
    });

    setProbabilityResult(probabilityResultArray)

    setAhpClicked(false)
    setTimeClicked(false)
    setProbabilityClicked(true)
}

useEffect(() => {

    const fetchRoutes = () => {
        try {
            const API_KEY =
process.env.REACT_APP_VISICOM_API_KEY;
            const destination = [incident.objectlongitude,
incident.objectlatitude]

            patrols.map(async patrol => {

```

```

        const origin = [patrol.longitude,
patrol.latitude]
        var probability;
        const response = await
axios.get("https://api.visicom.ua/data-
api/5.0/core/distance.json?origin=" + origin + "&destination=" +
destination + "&geometry=path&mode=driving-shortest&key=" +
API_KEY);
        if (incident.incidentdanger === "Наднизький")
{
            probability =
patrol.patrolprobabilitysuperlow;
        }
        if (incident.incidentdanger === "Низький") {
            probability = patrol.patrolprobabilitylow;
        }
        if (incident.incidentdanger === "Середній") {
            probability =
patrol.patrolprobabilitymedium;
        }
        if (incident.incidentdanger === "Високий") {
            probability =
patrol.patrolprobabilityhigh;
        }
        const newRoute = { patrolid: patrol.id,
response: response.data, distance:
response.data.properties.distance, probability: probability,
danger: incident.incidentdanger }
        setRoutes(r => [...r, newRoute])
        console.log(newRoute)
    })

    } catch (error) {
        console.log(error);
    }
};

const updateCheckSwitch = async (id) => {
    try {
        const BECKEND_URL =
process.env.REACT_APP_BECKEND_URL;
        await axios.put(BECKEND_URL + "/incident-ischeck/"
+ id);
    } catch (error) {
        console.log(error);
    }
}

const fetchIncident = async () => {
    const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;
    const response = await axios.get(BECKEND_URL +
"/incidents/" + params.id);
    setIncidents(response.data[0])
}

```

```

        if (incident.incidentchecked === 0) {
            updateCheckSwitch(incident.incidentid)
        }
        return response.data;
    }

    const fetchAllPatrols = async () => {
        const BECKEND_URL = process.env.REACT_APP_BECKEND_URL;
        const response = await axios.get(BECKEND_URL +
"/patrols-name");
        setPatrols(response.data);
        setPatrol(response.data[0])
        return response.data;
    };

    const fetchAllStatuses = async () => {
        try {
            const BECKEND_URL =
process.env.REACT_APP_BECKEND_URL;
            const response = await axios.get(BECKEND_URL +
"/statuses");
            setStatuses(response.data);
            setStatus(response.data[0]);
        } catch (error) {
            console.log(error);
        }
    };

    const fetchOperation = async () => {
        try {
            const BECKEND_URL =
process.env.REACT_APP_BECKEND_URL;
            const response = await axios.get(BECKEND_URL +
"/last-operation/" + incident.incidentid);
            setOperaton(response.data[0]);
        } catch (error) {
            console.log(error);
        }
    };

    fetchAllPatrols()
        .then((data) => fetchIncident())
        .then((data) => fetchRoutes());

    fetchOperation();

    fetchAllStatuses();
}, [params.id, incident.incidentchecked, incident.incidentid,
incident.objectlatitude, incident.objectlongitude,
incident.danger]);

```

```

return (
  <div className='incident-item-page'>
    <Toaster toastOptions={{
      style: {
        background: '#fff6df',
        color: '#233044',
      },
    }} />
    <div className='incident-item-info'>
      <div className='incident-item-div'>
        <span id="incident-id-title">Номер інциденту:
</span>
        <span id="incident-
id">{params.id}</span><br></br>
        <span id="incident-datetime-title">Дата й час
інциденту: </span>
        <span id="incident-datetime">{incident.date}
{incident.time}</span><br></br>
        <span id="incident-zones-title">Зони
спрацювання: </span>
        <span id="incident-
zones">{incident.incidentzones}</span><br></br>
        <span id="incident-iterations-title">Кількість
повторних: </span>
        <span id="incident-
iterations">{incident.incidentiterations}</span><br></br>
        <span id="incident-danger-title">Рівень
загрози: </span>
        <span id="incident-
danger">{incident.incidentdanger}</span><br></br>
      </div>
      <div className='object-item-div'>
        <span id="object-id-title">Номер об'єкту:
</span>
        <span id="object-
id">{incident.objectid}</span><br></br>
        <span id="incident-category-title">Категорія
об'єкту: </span>
        <span id="incident-
category">{incident.objectcategory}</span><br></br>
        <span id="incident-coordinates-
title">Координати об'єкту: </span>
        <span id="incident-
coordinates">({Number(incident.objectlatitude).toFixed(2)},
{Number(incident.objectlongitude).toFixed(2)})</span><br></br>
        <span id="object-address-title">Адреса
об'єкту: </span>
        <span id="object-
address">{incident.objectstreet} {incident.objecthouse}</span>
        {incident.objectsection && <span id="object-
section">, під'їзд {incident.objectsection}</span>}

```

```

        {incident.objectfloor && <span id="object-
floor">, поверх {incident.objectfloor}</span>
        {incident.objectapartment && <span id="object-
apartment">, квартира {incident.objectapartment}</span>}
    </div>
    <div className='client-item-div'>
        <span id="client-id-title">Номер власника:
</span>
        <span id="client-
id">{incident.clientid}</span><br></br>
        <span id="client-name-title">ПІВ власника:
</span>
        <span id="client-
name">{incident.clientname}</span><br></br>
        <span id="client-address-title">Адреса
власника: </span>
        <span id="client-
address">{incident.clientaddress}</span><br></br>
        <span id="client-phone-title">Телефон
власника: </span>
        <span id="client-
phone">{incident.clientphone}</span><br></br>
        {incident.clientalternatephone && <span
id="client-alternatephone-title">Альтернативний телефон власника:
</span>}
        {incident.clientalternatephone && <span
id="client-alternatephone">{incident.clientalternatephone}</span>}
    </div>
</div>
<div className='incident-item-actions'>
    {incident.length !== 0 && <div
className="incident-actions-div">
        <form onSubmit={setNewStatus}
className="incident-actions-form">
            <div>
                <span className="title-span">Поточний
статус: </span>
                {operation &&
<span>{operation.statustitle}</span>}
                {!operation && <span>відсутній</span>}
                <br></br><span className="title-
span">Бригада: </span>
                {operation &&
<span>{operation.patrolid} - {operation.firstpatrolmanname},
{operation.secondpatrolmanname}</span>}
                {!operation && <span>відсутня</span>}
                <br></br><label className="title-span"
htmlFor='status-id'>Змінити статус: </label>
                <select>
                    {statuses.map(status =>
                        <option onClick={e =>
setStatus(status)} key={status.id} value={status.id}>

```

```

{status.id} -
{status.title}
        </option>
    })
    </select>
    <br></br><label className="title-span"
htmlFor='patrol-id'>Обрати бригаду: </label>
    <select>
        {patrols.map(patrol =>
            <option onClick={e =>
setPatrol(patrol)} key={patrol.id} value={patrol.id}>
                {patrol.id} -
{patrol.firstpatrolmanname}, {patrol.secondpatrolmanname}
            </option>
        )}
    </select>
</div>

    <div>
        <button type="submit">Додати</button>
    </div>
</form>
</div>}
<div className="ahp-div">
    <button onClick={timeClick}>Обрати найближчу
бригаду</button><br></br>
    <button onClick={probabilityClick}>Обрати
бригаду з кращим показником ймовірності своєчасного та правильного
виконання</button><br></br>
    <button onClick={calculateClick}>Обрати
оптимальну бригаду враховуючи обидва критерії</button><br></br>
    <button onClick={universalClick}>Обрати
оптимальну бригаду за довільними критеріями</button><br></br>

    {ahpResult.map(result => {
        if (ahpClicked) {
            return <div key={result.patrolid}>
                <span className='title-
span'>Бригада: </span>
                    <span>{result.patrolid}, </span>
                    <span className='title-
span'>Результат: </span>
                        <span>{result.score}, </span>
                        <span className='title-
span'>Відстань: </span>
                            <span>{result.distance}м, </span>
                            <span className='title-span'>Час:
</span>
                                <span>{result.time}с, </span>
                                <span className='title-
span'>Ймовірність: </span>

```





```

        } else {
            return <span
key={result.patrolid}></span>
        }
    }
})

</div>
</div>
<div className='incident-item-map'>

    <MapContainer
        dragging={scrollWheelZoom}
        center={coordinates} zoom={mapZoom}
        ref={mapRef}>
        <TileLayer
            attribution='Дані карт © AT «<a
href="https://api.visicom.ua/">Visicom</a>»'

url={"https://tms{s}.visicom.ua/2.0.0/ua/base/{z}/{x}/{y}.png?key=
" + VISICOM_API_KEY}
            subdomains={'123'}
            tms={true}
        />

        {routes.map(route =>
            route.patrolid === patrol.id && <GeoJSON
key={route.patrolid} data={route.response} />
        )}

        {patrols.map(patrol =>
            <PatrolMarker patrol={patrol}
key={patrol.id} />
        )}

        {incident.length !== 0 &&
            <Marker
position={[incident.objectlatitude,
incident.objectlongitude]}></Marker>}

    </MapContainer>

</div>
</div>
);
};

export default IncidentsItemPage;

```

## ДОДАТОК Д

Копії публікацій.



### Класифікатор ситуацій для підтримки прийняття рішень операторів централізованих пультів спостереження за об'єктами

Рудик В.В., студент; Лавров Є.А., професор  
Сумський державний університет, м. Суми, Україна

**Вступ.** Оператори централізованих пульта спостереження за об'єктами [1] працюють часто в умовах стресу і дефіциту часу. Для ефективної діяльності операторів необхідно впроваджувати системи підтримки прийняття рішень [2,3].

**Постановка задачі.** Для системи підтримки прийняття рішень розробити класифікатор ситуацій.

**Результати.** В якості методологічної бази дослідження обрано методологію машинного навчання та алгоритм побудови дерева рішень. Принцип побудови класифікатора розглянемо на прикладі визначення рівня загрози сигналу про спрацювання об'єкту, що приходить оператору централізованого пульта спостереження за об'єктами, в залежності від 4 показників: категорія об'єкту, кількість повторних спрацювань, кількість зон спрацювання та час, пройдений з моменту спрацювання. Фрагмент навчальної матриці наведено в табл. 1., а фрагмент реалізації – на рис. 1.

Таблиця 1. Визначення рівня загрози сигналу. Навчальна матриця.

Категорія об'єкту	Кількість повторних спрацювань	Кількість зон спрацювання	Пройдений час з 1 спрацювання	Рівень загрози
6	3	1	105	Низький
3	6	2	170	Низький
4	1	1	50	Фейк
1	5	4	30	Середній
2	10	4	60	Високий
1	3	8	86	Високий
5	3	1	50	Низький
1	5	5	96	Високий
2	3	2	25	Низький
3	7	5	98	Високий
4	11	2	82	Високий
5	5	2	50	Середній
6	6	2	60	Середній
2	4	4	80	Середній
1	3	1	30	Фейк
4	1	1	5	Фейк
3	2	7	40	Середній
1	7	4	80	Високий
6	0	1	100	Фейк
4	7	6	50	Середній
5	12	2	20	Низький
3	3	3	104	Середній
2	6	2	30	Низький
5	10	5	70	Високий

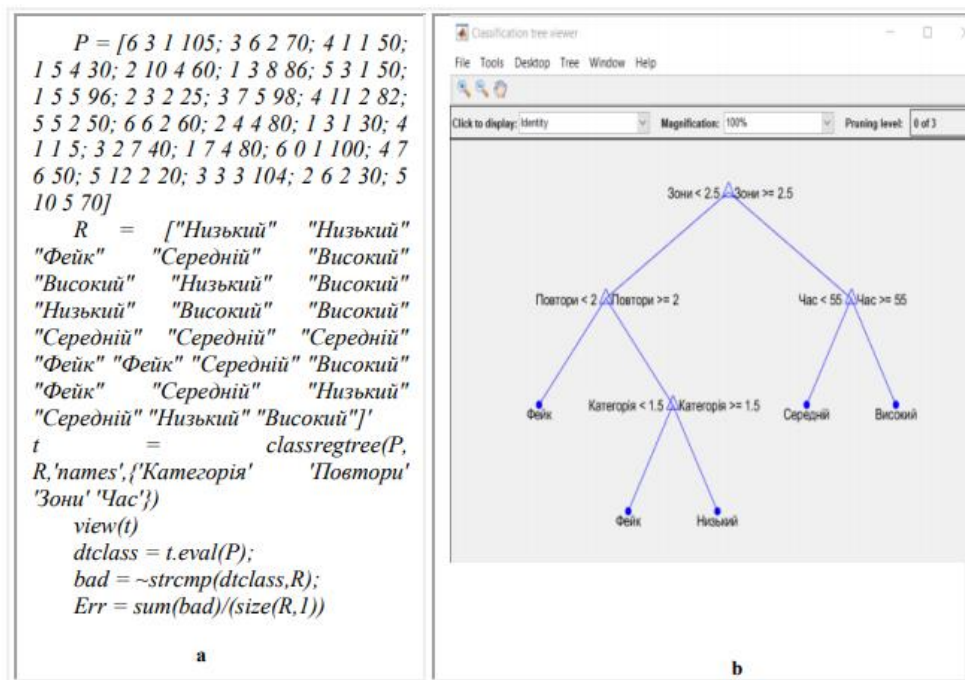


Рисунок 1 – Класифікатор “рівень загрози”:

а – Фрагмент реалізації (matlab),  
 б – дерево рішень (фрагмент)

**Висновки.** Класифікатор є зручним елементом системи підтримки прийняття рішень оператора централізованого пульта спостереження за об’єктами.

1. *Організація роботи централізованої охорони* [Електронний ресурс] – Режим доступу до ресурсу: [https://ohranatruda.ru/ot\\_biblio/norma/392518/](https://ohranatruda.ru/ot_biblio/norma/392518/)
2. E. Lavrov, P. Paderno, E. Burkov, A. Volosiuk, V.D. Lung, *E3S Web of Conferences. EDP Sciences*. **166** (2020).
3. E. Lavrov, N. Pasko, O. Siryk, V. Mukoseev, S. Dubovyk, *CEUR Workshop Proceedings. CEUR-WS*. **2740**, 357 (2020).

### **Моделювання діяльності людини-оператора централізованого пульта спостереження за об'єктами**

Рудик В.В., студент; Лавров Є.А., професор., Пасько Н.Б., доцент  
Сумський державний університет, м. Суми, Україна

**Вступ.** Оператори централізованих пульта спостереження за об'єктами є одними з найголовніших підрозділів в роботі спеціалізованих установ, покликаних реагувати на екстремальні ситуації, що відбуваються в приватних квартирах, будинках і організаціях. Через наявність багатьох негативних факторів їх робота може бути неточною, повільною, неефективною, а іноді й помилковою [1]. Моделювання діяльності оператора, реалізація автоматизованих процедур оцінювання показників діяльності допоможуть варіювати значеннями факторів, що впливають, зменшити шанс на помилку оператора при виконанні операцій, забезпечивши таким чином задану ефективність роботи і умови праці на робочому місці.

**Постановка задачі.** Для системи підтримки прийняття рішень розробити систему варіантного моделювання, дає можливість розраховувати: вірогідності безпомилкового виконання діяльності, математичного очікування та дисперсії часу виконання діяльності та вірогідності своєчасного виконання діяльності в залежності від структур діяльності та значень факторів, що впливають на оператора.

**Результати.** В якості методологічної бази дослідження обрано функціонально-структурну теорію професорів А. І. Губінського, В.Г. Євграфова, А.Т. Ашерова, П.П. Чабаненко, П.П. Падерно, Є. А. Павлова [2-4] і розвинений цією науковою школою метод функціональних мереж.

Для досягнення мети було обстежено робочі місця (рис.1.а) та діяльність операторів, побудовано моделі алгоритмів (приклад моделі – рис. 2), сформовано дані про час і безпомилковість виконання оператором окремих операцій, розроблено математичні моделі для типових функціональних структур діяльності оператора, розроблено програмне забезпечення для оцінки діяльності операторів, оцінено альтернативні варіанти діяльності, розроблено рекомендації по організації діяльності операторів.



Рисунок 1 – Робоче місце оператора централізованого пульта спостереження (а) та фрагмент моделі структури діяльності по управлінню мобільною бригадою (b).

**Висновки.** Розроблена інформаційна система моделювання діяльності людини-оператора централізованого пульта спостереження за об'єктами дозволяє проаналізувати алгоритм роботи оператора, коригувати значення показників діяльності, розрахувати помилковість виконання операцій та видати рекомендації щодо покращення алгоритму діяльності оператора.

1. *Організація роботи централізованої охорони* [Електронний ресурс] – Режим доступу до ресурсу: [https://ohranatruda.ru/ot\\_biblio/norma/392518/](https://ohranatruda.ru/ot_biblio/norma/392518/)
2. А.Н. Адаменко, А.Т. Ашерев, Е.А. Лавров, и др., *Информационно-управляющие человеко-машинные системы: Исследование, проектирование, испытания: Справочник* (М.: Машиностроение: 1993).
3. E. Lavrov, O. Siryk, P. Chabanenko, *CEUR Workshop Proceedings. CEUR-WS 2732*, 572 (2020).
4. E. Lavrov, N. Pasko, O. Siryk, V. Mukoseev, S. Dubovyk, *CEUR Workshop Proceedings. CEUR-WS. 2740*, 357 (2020).



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
МІЖНАРОДНИЙ УНІВЕРСИТЕТ «АСТАНА»**



# **ІНФОРМАТИКА, МАТЕМАТИКА, АВТОМАТИКА**

**ІМА - 2023**

**МАТЕРІАЛИ  
та програма**

**МІЖНАРОДНОЇ  
НАУКОВОЇ КОНФЕРЕНЦІЇ  
МОЛОДИХ УЧЕНИХ**

**(Суми-Астана,  
24-28 квітня 2023 року)**

**Суми,  
Сумський державний університет  
2023**



**Людино-системний підхід до вирішення  
задач забезпечення надійності обробки інформації  
в системах автоматизованого проектування і управління**

Лавров Є.А., *професор*; Рудик В.В., *студент IT.м-11н*;  
Клименко І.В., *аспірант*; Солонар І.О., *аспірант*; Чечель О.В., *аспірант*  
Сумський державний університет, м. Суми, Україна

**Вступ.** Статистика подій та аварій на складних об'єктах показує, що понад 80% випадків пов'язано з людським фактором [1-3].

**Постановка задачі.** Розробити методологію забезпечення надійності обробки інформації в системах автоматизованого проектування і управління засобами врахування “людського фактору” та методів ергономічного забезпечення.

**Результати.** Проблема запобігання помилок у різних автоматизованих системах (АС) докладно розглядалась у роботах А.Т. Ашерова (системи комплексної автоматизації), Г.М. Зараковського, Ю.К. Стрелкова та В.А. Бодрова (в авіації), О.М. Анохіна, А.Г. Чачко та В.М. Абрамової (в атомній енергетиці), А.І. Губінського та В.Г. Євграфова (морський флот), Б.Ф. Ломова та інших авторів. Помилки можуть відбуватися з вини людини-оператора внаслідок неякісної розробки системи або поганих умов експлуатації. Існує безліч методів та засобів, які спрямовані на зниження ймовірності помилок людини-оператора, що здійснюються під час вирішення завдань управління технологічним об'єктом або проектування систем різноманітного призначення. Аналіз більшості проблем свідчить про відсутність системного підходу до проектування людино-машинної взаємодії. В зв'язку з цим пропонуємо нову парадигму, основу на запропонованій професором Анатолієм Губінським концепції, яка орієнтується на кардинально новий підхід до проектування АС – людино-системний підхід [2]. Цей підхід призводить до зміни існуючої методології проектування. Щоб забезпечити гуманізацію роботи людини та підвищення ефективності АС за рахунок врахування людського фактора, потрібно розглядати систему з позицій людино-системного підходу. Класичний системний підхід до

дослідження складних систем (наприклад, у системотехніці) полягає в тому, що вона розглядається з урахуванням всіх взаємодіючих елементів і описується у вигляді «абстрактної (знакової) системи» (етап системного аналізу), працюючи з якою, розробник АС може вибрати найбільш раціональні організаційно-проектні рішення (етап системного синтезу).

При людино-системному підході зберігається ця методологія, але підкреслюються обов'язкове врахування людського фактору та центральна роль людини-оператора[2]. Мета системного аналізу – виявлення складу компонентів АС та взаємозв'язків між ними та формалізоване (для однозначного розуміння) подання цієї інформації у вигляді багатовимірної абстрактної системи, яка називається «інтегральна модель (ІМ) АС». Мета системного синтезу – шляхом оперування з ІМ АС обґрунтувати, як необхідно організувати діяльність людини (без перевантажень і недовантажень) і які ергономічні заходи повинні бути проведені, щоб система в цілому була ефективна, гуманна, безпечна, безаварійна та екологічна. В якості базової моделі для нової концепції людино-системного підходу пропонуємо модель керованої функціональної мережі [3]. Підхід апробовано при проектуванні систем забезпечення безпеки критичних об'єктів.

1. E. A Lavrov et al . Analysis of information security issues in corporate computer networks 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1047 012117
2. E. Lavrov, N. Pasko, O. Siryk, P. Paderno and E. Burkov, "Models and Information Technology for Reliable Design of the Functioning Processes of Flexible Production Systems as Complex Human-Machine Systems," 2020 IEEE Int. Conf. PIC S&T, 2020, pp. 673-678, doi: 10.1109/PICST51311.2020.9467894.
3. E. Lavrov et al. The Methodology of Managed Functional Networks for Organizing Effective and Adaptive Human-Machine Dialogue in Automated Systems 2021 CEUR Workshop Proceedings 2021 3013, p. 428-437

### **Інформаційна технологія підтримки прийняття рішень при управлінні безпекою складних об'єктів**

*Рудик В.В., студент ІТ.м-11н; Лавров Є.А., професор  
Сумський державний університет, м. Суми, Україна*

**Вступ.** Оператори спеціальних підрозділів охорони складних об'єктів керують процесами усунення проблемних ситуацій та управляють діяльністю оперативних мобільних бригад [1-3]. З метою забезпечення діяльності операторів розробляють спеціальні інформаційні системи, які повинні бути обладнані підсистемами підтримки прийняття рішень, побудованими за принципами варіантного аналізу і оптимізації діяльності охоронців[4].

**Постановка задачі.** Розробити інформаційну технологію, орієнтовану на: облік заявок і управління інцидентами в системах забезпечення безпеки складних об'єктів; моделювання варіантів призначення на реалізацію поточних інцидентів оперативних мобільних бригад і оптимізацію розподілу заявок між бригадами.

**Результати.** *Функціональна структура системи підтримки прийняття рішень.*

Запропоновано наступну функціональну модель системи: Підсистема (ПС) визначення рівня загрози; ПС аналізу можливості виїзду і часу транспортування бригад на місце події; ПС аналізу і прогнозу безпомилковості реалізації алгоритму діяльності по усуненню небезпеки; ПС вибору оптимального варіанту розподілу інцидентів; ПС обліку і документування інцидентів безпеки. *Моделі і засоби підтримки прийняття рішень.* Визначення рівня загрози – за принципом класифікатора, побудованого по методології “дерев рішень”. Аналіз оптимального маршруту – за допомогою API (application programming interface) картографічної платформи Visicom. Прогнозування безпомилковості і своєчасності реалізації алгоритмів діяльності по усуненню загроз – за методологією функціональних мереж на основі побудови алгоритмів діяльності і їх автоматичної редукції з використанням початкових даних, які характеризують ефективність виконання окремих дій різними мобільними бригадами (накопичуються в базі даних системи підтримки прийняття рішень). Вибір оптимального варіанту розподілу інцидентів – на основі постановки і вирішення задачі багатокритеріальної оптимізації з врахуванням в т.ч. показників : часу прибуття бригади на місце події; вірогідності безпомилкового усунення загрози; вірогідності своєчасного усунення загрози; інших (система допускає розширення системи показників). Система

оптимізації передбачає можливість отримання рішення з використанням сукупності різноманітних постановок задач. Реалізовані моделі лінійного програмування, а також моделі, основані на принципі методу аналізу ієрархій.

**Реалізація.** Систему реалізовано мовою JavaScript. Фрагмент інтерфейсу системи підтримки прийняття рішень представлено на рис 1.

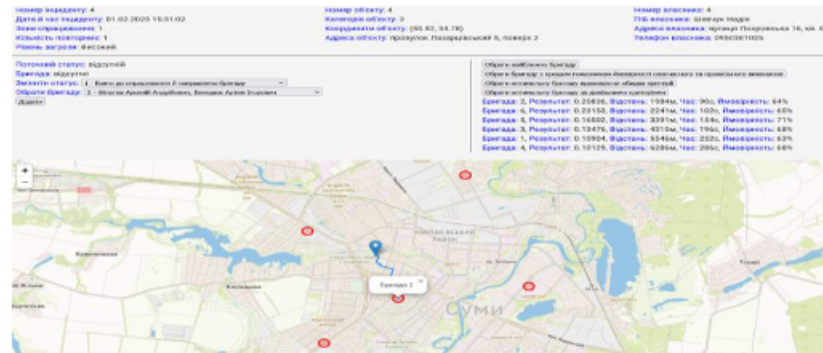


Рисунок 1 – Інтерфейс системи підтримки прийняття рішень

**Висновки.** Інформаційна технологія забезпечує прийняття рішень на основі об'єктивних кількісних показників та передбачає зменшення напруженості, а також когнітивний комфорт для операторів-керівників і сприяє підвищенню безпеки об'єктів, що охороняються.

1. Організація роботи централізованої охорони [Електронний ресурс] – Режим доступу до ресурсу: <http://parusconsultant.com/?doc=07Z5925E93>
2. Рудик В.В., Лавров Є.А., Пасько Н.Б. Моделювання діяльності людини-оператора централізованого пульта спостереження за об'єктами // Інформатика, Математика, Автоматика (ІМА – 2021), Суми–Нур-Султан, 19–23 квітня 2021 р, Суми, СумДУ, 2021, С. 109-110
3. Рудик В.В., Лавров Є.А. Класифікатор ситуацій для підтримки прийняття рішень операторів централізованих пультів спостереження за об'єктами// Інформатика, Математика, Автоматика (ІМА – 2021), Суми–Нур-Султан, 19–23 квітня 2021 р, Суми, СумДУ, 2021, С. 107-108
4. Lavrov, E. et. al. (2020). Expert assessment systems to support decision-making for sustainable development of complex technological and socioeconomic facilities. In E3S Web of Conferences (Vol. 166). EDP Sciences. <https://doi.org/10.1051/e3sconf/202016611002>.

### **Метод визначення рівня загрози для підтримки прийняття рішень операторів систем управління безпекою складних об'єктів**

*Рудик В.В., студент ІТ.м-11н; Лавров Є. А., професор  
Сумський державний університет, м. Суми, Україна*

**Вступ.** Останнім часом підвищились рівні небезпеки для різноманітних об'єктів інфраструктури населених пунктів та приватних господарств. В зв'язку з такими проблемами створюються спеціальні підрозділи управління безпекою[1-3]. Такі підрозділи мають бути обладнані системами підтримки прийняття рішень. У функції оператора, що керує процесами виявлення і усунення проблемних ситуацій, входить прийняття рішень про розподіл задач між оперативними мобільними бригадами та контроль ефективності вирішення задач. Однією з проблемних точок є визначення ступеню критичності ситуацій, який суттєво впливає на засоби вирішення задач. В роботі [3] поставлено задачу розробки класифікатора ситуацій і продемонстровано можливість побудови такого класифікатора.

**Постановка задачі.** В зв'язку з виявленими проблемними питаннями застосування на практиці класифікатора [5] ставимо завдання :провести дослідження з метою підвищення ефективності навчання класифікатора з врахуванням актуальних факторів, що впливають на формування значення вихідного класу; реалізувати класифікатор ситуацій.

**Результати.** Проведені дослідження ефективності класифікатора [5] виявили проблему низької ефективності – відсутність в моделі входу типу “*наявність об'єктів, що створюють шуми*”. В зв'язку з цим модернізовано методику побудови навчальної матриці і саму модель класифікатора.

Фрагмент навчальної матриці наведено в табл.1., а фрагмент реалізації – на рис.1.

**Висновки.** Метод визначення рівня загрози для підтримки прийняття рішень операторів систем управління безпекою складних об'єктів, побудований на основі методології навчання з учителем і моделі типу “*дерево рішень*”, може бути ефективним інструментом для класифікації ситуацій, що виникають при спостереженні за складними об'єктами.

Таблиця 1. Визначення рівня загрози безпеці об'єкту. Навчальна матриця (фрагмент).

Категорія об'єкту	Кількість зон спрацювання	Кількість повторних спрацювань	Назвність об'єктів, що створюють шуми	Рівень загрози
1	5	8	0	Високий
2	1	4	0	Середній
3	2	2	1	Низький
3	1	0	1	Наднизький
2	4	6	1	Високий
1	1	0	0	Низький
3	1	6	1	Середній
2	1	1	1	Низький
3	5	5	0	Високий
2	1	2	0	Середній
3	3	0	1	Низький
1	3	0	0	Високий
2	1	0	1	Наднизький
2	2	1	1	Низький
1	1	0	0	Наднизький
2	2	0	1	Наднизький
2	3	0	0	Середній
2	3	1	1	Низький
3	2	1	1	Низький
1	2	1	0	Високий
3	2	1	0	Середній
2	1	1	0	Середній
3	1	1	0	Низький

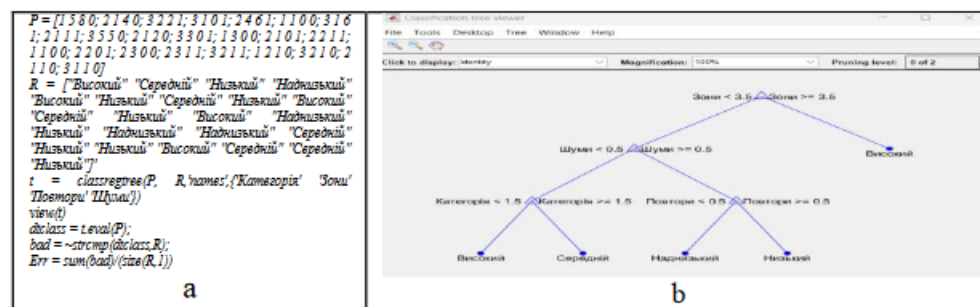


Рисунок 1 – Класифікатор “рівень загрози”:

а – Фрагмент реалізації (matlab), б – дерево рішень (фрагмент)

- 1 Організація роботи централізованої охорони [Електронний ресурс] – Режим доступу до ресурсу: <http://parusconsultant.com/?doc=07Z5925E93>
- 2 Рудик В.В., Лавров Є.А., Пасько Н.Б. *Інформатика, Математика, Автоматика (ІМА – 2021), Суми–Нур-Султан, 19–23 квітня 2021 р, Суми, СумДУ, 2021, С. 109-110*
- 3 Рудик В.В., Лавров Є.А. Класифікатор ситуацій для підтримки прийняття рішень операторів централізованих пультів спостереження за об'єктами//Інформатика, Математика, Автоматика (ІМА – 2021), Суми–Нур-Султан, 19–23 квітня 2021 р, Суми, СумДУ, 2021, С. 107-10

## ДОДАТОК Е



**АКТ**  
**Впровадження у робочий процес**  
**головного пункту централізованого спостереження ЦТО УПО в Сумській області**  
**результатів дипломної роботи**

студента групи ІТ-71-9 Рудик Віталія Віталійовича

на тему:

«Моделювання діяльності оператора централізованого пульта спостереження за об'єктами»

Складений 10 червня 2021 р. комісією у складі:

**Голова комісії:**

*Старший інженер ГПЦС ЦТО Замотайло В.О.*

**Члени комісії:**

*1. Інженер ГПЦС ЦТО Курако В.Ю.*

*2. Інженер ГПЦС ЦТО Ромащенко Л.І.*

В період з 01 червня 2021 р. по 09 червня 2021 р. комісія провела роботу з визначення впровадження результатів дослідження студента у робочий процес головного пункту централізованого спостереження ЦТО УПО в Сумській області.

### Результати роботи комісії

1. На головний пункт централізованого спостереження ЦТО УПО в Сумській області передано інформаційну технологію моделювання діяльності оператора централізованого пульта спостереження за об'єктами.

2. Результати роботи використовуються організацією для визначення часових та надійнісних показників алгоритмів діяльності оператора та вибору раціональних алгоритмів.

Голова комісії

Члени комісії

 Замотайло В.О.  
 Курако В.Ю.  
 Ромащенко Л.І.