МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ,
ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ ТА РАДІОЕЛЕКТРОННИХ СИСТЕМ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Віктор ГНАТЮК
“_____”_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР

Тема: «Система моніторингу та оцінки загроз інформаційної безпеки технології Smart Home»

Виконавець: _____ Михайло БОГДАНЧИК
(підпис)

Керівник: _____ Ірина КОЗЛЮК
(підпис)

Консультанти з окремих розділів пояснювальної записки:

Консультант розділу «Охорона праці» _____ Батир ХАЛМУРАДОВ
(підпис)

Консультант розділу «Охорона навколишнього середовища»
_____ Андріан ЯВНЮК
(підпис)

Нормоконтролер: _____ Денис БАХТІЯРОВ
(підпис)

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації, електроніки та телекомунікацій
Кафедра телекомунікаційних та радіоелектронних систем
Спеціальність 172 «Телекомунікації та радіотехніка»
Освітньо-професійна програма «Телекомунікаційні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Віктор ГНАТЮК
“_____”_____2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Богданчика Михайла Павловича
(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема кваліфікаційної роботи: «Система моніторингу та оцінки загроз інформаційної безпеки технології Smart Home»
затверджена наказом ректора від «28» вересня 2023 р. №1965/ст
2. Термін виконання роботи: з 02.10.2023 р. по 31.12.2023 р.
3. Вихідні дані до роботи: технологія Smart Home
4. Зміст пояснювальної записки: огляд технології Smart Home; класифікація загроз інформаційній безпеці системи Smart Home; проєктування системи інформаційної безпеки; система моніторингу та оцінки загроз інформаційної безпеки технології Smart Home
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: XML Schema-файл для XML-файлу складу системи; XML Schema-файл для XML-файлу "ідеального" стану системи; XML Schema-файл для XML-файлу загроз "розумного бshинку"; лістинг - метод отримання "ідеального" стану; лістинг - метод додавання елемента лістинг - метод додавання елемента; лістинг - метод моніторингу даних

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Розробити деталізований зміст розділів кваліфікаційної роботи	02.10.2023-04.10.2023	Виконано
2	Вступ	05.10.2023-08.10.2023	Виконано
3	Огляд технології Smart Home	09.10.2023-22.10.2023	Виконано
4	Класифікація загроз інформаційній безпеці системи Smart Home	23.10.2023-05.11.2023	Виконано
5	Проєктування системи інформаційної безпеки	06.11.2023-14.11.2023	Виконано
6	Система моніторингу та оцінки загроз інформаційної безпеки технології Smart Home	15.11.2023-30.11.2023	Виконано
7	Охорона праці	01.12.2023-06.12.2023	Виконано
8	Охорона навколишнього середовища	07.12.2023-17.12.2023	Виконано
9	Усунення недоліків та захист кваліфікаційної роботи	18.12.2023-31.12.2023	Виконано

7. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	к.м.н., професор Батир ХАЛМУРАДОВ		
Охорона навколишнього середовища	к.б.н., доц. Андріан ЯВНЮК		

8. Дата видачі завдання: “22" вересня 2023 р.

Керівник кваліфікаційної роботи _____ Ірина КОЗЛЮК
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Михайло БОГДАНЧИК
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Кваліфікаційна робота «Система моніторингу та оцінки загроз інформаційної безпеки технології Smart Home» містить 85 сторінок, 20 рисунків, 5 таблиць, 60 використаних джерел.

СИСТЕМА МОНІТОРИНГУ, ОЦІНКА ЗАГРОЗ, ІНФОРМАЦІЙНА БЕЗПЕКА, SMART HOME, СХЕМА ЗАХИСТУ, КІБЕРБЕЗПЕКА, ВРАЗЛИВОСТІ, ПРОТОКОЛИ БЕЗПЕКИ, АУТЕНТИФІКАЦІЯ, ШИФРУВАННЯ, ЗАХИСТ ВІД КІБЕРАТАК, ВІДДАЛЕНИЙ МОНІТОРИНГ, ІДЕНТИФІКАЦІЯ КОРИСТУВАЧІВ, МЕРЕЖЕВА БЕЗПЕКА, ІНТЕРНЕТ РЕЧЕЙ, ДОСТУП ДО СИСТЕМИ, ІНФОРМАЦІЙНА АРХІТЕКТУРА, МОНІТОРИНГ ЗЛОВМИСНОГО ВИКОРИСТАННЯ, ЗАХИСТ ОСОБИСТИХ ДАНИХ, АНАЛІЗ ПОТЕНЦІЙНИХ ЗАГРОЗ.

Мета роботи - проєктування інформаційної системи моніторингу та оцінки загроз інформаційній безпеці технології "Smart Home" і розробка прототипу спроєктованої системи.

Об'єкт дослідження: технологія "Smart Home" і загрози інформаційній безпеці, що виникають у ній.

Предмет дослідження: методи проєктування системи інформаційної безпеки технології "Smart Home".

Практична значущість результатів - спроєктовану систему інформаційної безпеки технології "Smart Home" може бути використано для забезпечення інформаційної безпеки в будь-яких системах SH.

Результатом роботи є спроєктована система інформаційної безпеки технології УД і розроблений прототип системи.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ

IoT (Internet of Things) - Інтернет речей.

SIEM (Security Information and Event Management) - Система управління інформаційною безпекою та аналізу подій.

IDS (Intrusion Detection System) - Система виявлення вторгнень.

IPS (Intrusion Prevention System) - Система запобігання вторгненням.

CVE (Common Vulnerabilities and Exposures) - Загально відомі вразливості та вразливості експозиції.

DDoS (Distributed Denial of Service) - Розподілений відмова в обслуговуванні.

VPN (Virtual Private Network) - Віртуальна приватна мережа.

SSL (Secure Sockets Layer) - Безпечний протокол передачі даних.

DNS (Domain Name System) - Система доменних імен.

WPA (Wi-Fi Protected Access) - Захищений доступ до Wi-Fi.

ACL (Access Control List) - Список керування доступом.

EDR (Endpoint Detection and Response) - Виявлення та реагування на загрози на кінцевих пристроях.

MFA (Multi-Factor Authentication) - Багатофакторна аутентифікація.

PKI (Public Key Infrastructure) - Інфраструктура відкритих ключів.

SOC (Security Operations Center) - Центр операцій з безпеки.

UEBA (User and Entity Behavior Analytics) - Аналіз поведінки користувачів та об'єктів.

RTO (Recovery Time Objective) - Об'єктивний час відновлення.

RPO (Recovery Point Objective) - Об'єктивний час відновлення до точки у часі.

GDPR (General Data Protection Regulation) - Загальний регламент з захисту особистих даних.

COI (Conflict of Interest) - Конфлікт інтересів.

AI (Artificial Intelligence) - Штучний інтелект.

ML (Machine Learning) - Машинне навчання.

NIST (National Institute of Standards and Technology) - Національний інститут стандартів і технологій.

AES (Advanced Encryption Standard) - Розширений стандарт шифрування.

API (Application Programming Interface) - Інтерфейс програмування додатків.

OS (Operating System) - Операційна система.

XSS (Cross-Site Scripting) - Міжсайтовий скриптинг.

CSRF (Cross-Site Request Forgery) - Міжсайтовий підроблення запитів.

SSL/TLS (Transport Layer Security) - Безпека транспортного рівня.

LAN (Local Area Network) - Локальна мережа.

WAN (Wide Area Network) - Широкосмугова мережа.

OTA (Over-the-Air) - Оновлення через повітря.

PII (Personally Identifiable Information) - Особиста ідентифікаційна інформація.

SOCaaS (Security Operations Center as a Service) - Центр операцій з безпеки як сервіс.

UTM (Unified Threat Management) - Об'єднане управління загрозами.

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) - Повністю автоматизований тест Тьюрінга для відрізнення комп'ютерів і людей.

SIM (Subscriber Identity Module) - Картка абонента.

IoTaaS (IoT as a Service) - Інтернет речей як сервіс.

FOTA (Firmware Over-The-Air) - Прошивка через повітря.

MAC (Media Access Control) - Контроль доступу до медіа.

ВСТУП

Актуальність теми. В сучасному світі технологічний прогрес надзвичайно швидко трансформує наше повсякденне життя, впливаючи на всі аспекти суспільства. Однією з важливих тенденцій цього розвитку є впровадження технологій "розумних будинків" (Smart Home), які дозволяють автоматизувати та оптимізувати обслуговування приміщень, підвищуючи комфорт та ефективність використання ресурсів.

Разом з цим, зростає ймовірність знаходження інформаційних систем та пристроїв "Smart Home" під загрозою кібератак, які можуть спричинити значні майнові та особисті втрати. Однією з ключових аспектів забезпечення надійності та безпеки використання таких технологій є розробка та впровадження системи моніторингу та оцінки загроз інформаційної безпеки технології "Smart Home".

Кваліфікаційна робота присвячена вивченню, аналізу та розробці ефективної системи моніторингу та оцінки ризиків безпеки для технології "Smart Home". Вона спрямована на створення комплексного підходу до захисту інформаційних ресурсів та персональних даних, що зберігаються та оброблюються в системах "розумних будинків". Акцент роботи буде зроблено на розробці методів виявлення, аналізу та управління потенційними загрозами та вразливостями, а також на розробці рекомендацій щодо забезпечення надійності та безпеки використання технології Smart Home.

Система "Smart Home" є апаратно-програмним комплексом, усередині якого обробляється великий потік інформації, у зв'язку з чим система "Smart Home" схильна до загроз інформаційній безпеці. На жаль, сучасні розробки в галузі технології SH не містять єдиної методології опису систем SH, тому відсутня і єдина методологія виявлення та оцінки загроз інформаційній безпеці технології "Smart Home".

Тестування декількох загальнодоступних систем SH, проведене компанією AV-TEST, доводить наявність проблем з інформаційною безпекою в пропонованих компаніями системах SH. Багато наявних рішень для додаткового захисту інформаційної безпеки SH використовують метод роботи, що полягає в під'єднанні до роутера і мо-

ніторингу потоку інформації між під'єднаними до Wi-Fi пристроями. Цей метод обмежує коло підтримуваних пристроїв. Також деякі з наявних рішень здійснюють додатковий збір і надсилання даних про роботу пристроїв ПД у хмарне сховище, що може стати додатковою загрозою [1-43]. Наведені аспекти аналізу стану галузі інформаційної безпеки технології ПД доводять актуальність розглянутої теми.

Мета роботи - проєктування інформаційної системи моніторингу та оцінки загроз інформаційній безпеці технології "Smart Home" і розробка прототипу спроєктованої системи.

Об'єкт дослідження: технологія "Smart Home" і загрози інформаційній безпеці, що виникають у ній.

Предмет дослідження: методи проєктування системи інформаційної безпеки технології "Smart Home".

Дослідження з проблематики системи моніторингу та оцінки загроз інформаційної безпеки технології Smart Home вимагає використання комплексу **наукових та практичних методів**. Нижче наведено основні методи, які використовувались у дослідженні цієї теми:

- **Літературний аналіз та огляд наукових джерел.** Огляд і аналіз наукових статей, книг, патентів, наукових конференцій та інших публікацій, пов'язаних з інформаційною безпекою та технологією Smart Home дозволяє отримати загальне розуміння сучасного стану проблеми та існуючих методів розробки систем безпеки.
- **Аналіз систем безпеки технології Smart Home.** Глибокий аналіз існуючих систем безпеки технології Smart Home, включаючи їхню архітектуру, алгоритми шифрування, методи аутентифікації та авторизації. Дослідження слабких місць і вразливостей існуючих систем дозволить виявити потенційні загрози.
- **Емпіричні дослідження та експерименти.** Проведення практичних тестів та експериментів на реальних пристроях Smart Home з метою оцінки рівня захищеності від різних кіберзагроз.

- **Експертні оцінки фахівців.** Отримання оцінок та консультацій від експертів у галузі інформаційної безпеки та Smart Home для з'ясування найбільш актуальних аспектів та можливих шляхів вдосконалення безпеки.
- **Моделювання та аналіз загроз.** Створення математичних моделей для аналізу можливих загроз та їхнього впливу на технологію Smart Home. Моделювання дозволяє оцінити ризики та розробити ефективні заходи безпеки.
- **Статистичний аналіз інцидентів.** Аналіз статистичних даних про кіберінциденти, пов'язані з технологією Smart Home, для визначення тенденцій та основних векторів атак.

Практична і наукова новизна: розробка класифікації вразливостей і загроз системи SH, що базується на зв'язку об'єкта управління і загрози, проєктування і розробка прототипу системи інформаційної безпеки, що налаштовується під конкретну систему SH.

Результатом роботи є спроєктована система інформаційної безпеки технології SH і розроблений прототип системи.

Практичне значення отриманих результатів та розробленої комп'ютерної програми у контексті дипломної роботи "Система моніторингу та оцінки загроз інформаційної безпеки технології Smart Home" надзвичайно важливе для сфери інформаційної безпеки та розробників технологій Smart Home. Основні аспекти його значущості наступні:

- **Забезпечення безпеки користувачів Smart Home.** Розроблена комп'ютерна програма дозволить здійснювати постійний моніторинг та оцінку рівня безпеки технології Smart Home для кінцевих користувачів. Це важливо для запобігання можливим кібератакам та недозволеним доступам до особистої інформації.
- **Ефективна реакція на потенційні загрози.** Система моніторингу та оцінки, впроваджена через розроблену програму, дозволить оперативно виявляти та аналізувати загрози безпеці Smart Home та вживати відповідні заходи для їх нейтралізації.

- **Захист особистої інформації та конфіденційності даних.** Розроблена програма сприятиме забезпеченню конфіденційності особистих даних користувачів, які зберігаються та оброблюються в системах Smart Home. Це важливо для відчуття спокою та безпеки користувачами.
- **Підвищення довіри до технологій Smart Home.** Враховуючи загрози кібербезпеки, програма та отримані результати сприятимуть підвищенню довіри споживачів до технологій Smart Home, що є важливим фактором для їх широкого прийняття та успішного розвитку.
- **Покращення виробничих процесів та розробок.** Отримані результати можуть бути використані розробниками для вдосконалення систем безпеки вже існуючих продуктів Smart Home та для розробки нових, більш захищених та надійних рішень.

РОЗДІЛ 1
ОГЛЯД ТЕХНОЛОГІЇ SMART HOME

1.1. Розумний будинок

Під терміном "Smart Home" (SH) заведено розуміти сукупність під'єднаних у загальну мережу пристроїв, що виконують певні дії з мінімальною участю людини. Ідея створення SH у наближеному до сьогоднішнього розуміння цього терміна з'явилася наприкінці 20 століття. Один із перших таких будинків було описано в журналі "Popular Mechanics" у 1950 році [3]. Термін "розумний будинок" було введено в 1984 році американською Асоціацією житлово-будівельних компаній [4], і до 2000 року ідея "розумних будинків" мала достатнє поширення в Європі і, особливо, в США. В Україні перші згадки про існування технології домашньої автоматизації з'явилися лише близько 1990 року [5, 6].

Технологія SH використовується для різних цілей, можна виділити основні з них [7]:

- тепло- та енергозбереження;
- підвищення комфорту;
- забезпечення безпеки.

Різні підприємства застосовують технологію SH в основному для енергозбереження та безпеки, використання технології для житлових приміщень може бути для всіх вищеописаних цілей.

Компанії-виробники систем SH пропонують різні варіації систем: готові рішення "під ключ" і ті, що налаштовуються під вимоги конкретного клієнта. Також на ринку представлені окремі "розумні" продукти (в основному випускаються виробниками техніки), які користувач може самостійно об'єднати в систему SH.

Системи "Smart Home" не мають єдиної методології опису. Як компанії, так і дослідники технології SH, мають різні підходи до опису системи SH. Підхід, що зустрічається найчастіше, - поділ системи SH на різні підсистеми. Узагальнений опис видів підсистем SH представлено на рисунку 1.1.

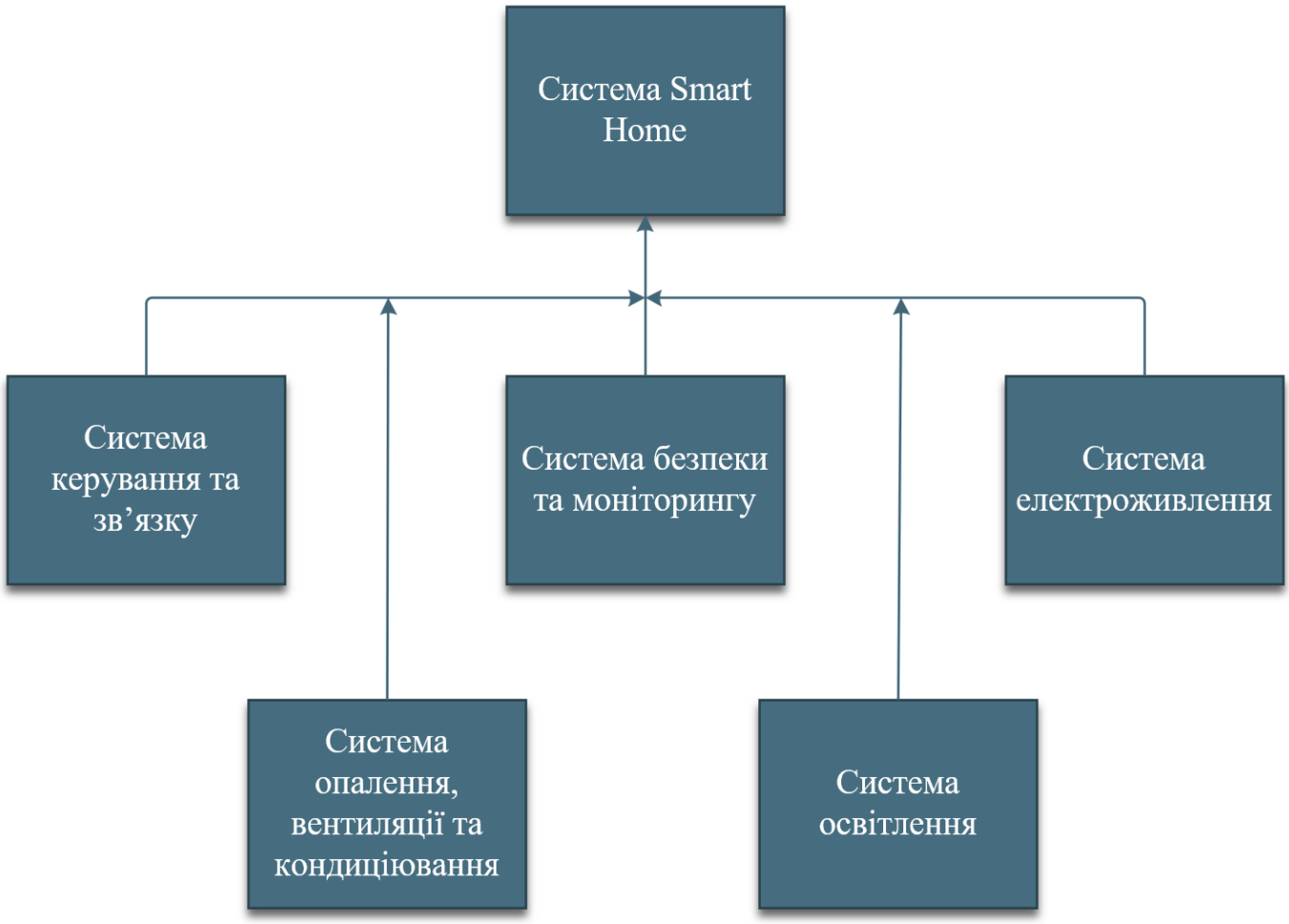

Рис. 1.1. Підсистеми "розумного будинку"

1.2. Захист інформаційної безпеки

Система SH є об'єктом інформатизації, схильним до загроз інформаційній безпеці [8]. Загрози інформаційній безпеці системи залежать від методів побудови системи, технологій, що використовуються, та інформаційних потоків, що обробляються [1], тому не існує єдиної методології способу захисту інформаційної безпеки.

Багато систем SH мають вбудований компонент захисту інформаційної безпеки, але, на жаль, цей компонент не завжди здійснює високий рівень захисту. У деяких системах компонент інформаційної безпеки може навіть бути відсутнім. З цієї причини деякі компанії почали виробництво додаткових засобів захисту інформаційної безпеки технології SH, але, якщо компанія є виробником "розумних" пристроїв, то в більшості випадків засіб захисту, який розробляється, здатний працювати тільки з "розумними пристроями" цієї компанії.

У 2014 році компанія AV-TEST, що є незалежним інститутом інформаційної безпеки, провела тестування [2] кількох загальнодоступних систем SH. AV-TEST досліджувала:

- наявність шифрованого зв'язку між елементами SH;
- використання активної аутентифікації;
- можливість зовнішньої маніпуляції;
- рівень захищеності віддаленого доступу. Результати тестування семи систем SH показали:
- лише три системи забезпечують інформаційну безпеку;
- дві системи недостатньо захищені і можуть бути схильні до внутрішніх атак;
- дві системи мають дуже слабкий інформаційний захист і можуть бути схильні як до внутрішніх, так і до зовнішніх атак.

Проведене тестування доводить, що навіть користувачам готових рішень не слід забувати про інформаційний захист і за потреби використовувати додаткові засоби захисту, а виробникам SH потрібно випускати продукти з вищим рівнем захисту.

1.3. Існуючі рішення захисту інформації систем "Smart Home"

Як було сказано раніше, компанії-виробники засобів захисту інформації для SH, які одночасно є виробниками "розумних" пристроїв, найчастіше розробляють засоби захисту для своїх пристроїв. Але інші компанії виробляють універсальні засоби захисту інформації. Далі розглянуто кілька прикладів таких пристроїв.

Один із поширених методів роботи на ринку пристроїв додаткового інформаційного захисту SH - під'єднання до роутера і моніторинг потоку інформації між під'єднаними до Wi-Fi пристроями.

Три яскраві приклади подібних пристроїв:

- Dojo, розроблене невеликою ізраїльською компанією Dojo-Labs [9];
- CUJO, розроблене групою каліфорнійських дослідників [10];
- Bitdefender Box, вироблене великою компанією Bitdefender [11].

Основний недолік цих пристроїв зумовлений вибором методу роботи, що обмежує пристрої, які захищаються. Так само пристрої Dojo і CUJO здійснюють додатковий збір і надсилання в "хмару" даних про роботу пристроїв, для визначення нових загроз. Додатковий збір даних може бути додатковою загрозою інформаційній безпеці, оскільки віддалена "хмара" може бути так само схильною до атаки хакерів.

Існують також і більш функціональні засоби захисту інформаційної безпеки, які можуть контролювати всю мережу SH і не збирають додаткові дані. Приклад подібних пристроїв - серія пристроїв Cisco ASA 5500-X з функціями FirePOWER [12], вироблена однією з найбільших IT-компаній Cisco. Але пропоновані функціональність і якість мають відповідну ціну і складність встановлення та експлуатації, оскільки подібні пристрої розроблено в основному для використання в середніх і великих компаніях, які мають необхідний персонал або можливість використання досить дорогого сервісного обслуговування.

1.4. Модель побудови системи моніторингу інформаційної безпеки

Відсутність єдиної методології побудови захисту інформаційної безпеки технології SH пояснює різні структури та методи роботи засобів захисту. Розробку цих засобів можна умовно розділити на такі етапи:

- опис моделі системи SH;
- опис моделі загроз інформаційної безпеки;
- розроблення методів оцінювання загроз;

- розробка автоматичного механізму моніторингу стану захисту SH.

Методи оцінювання загроз ґрунтуються на моделі системи SH і моделі загроз, можуть використовувати список найвірогідніших загроз і критерії: джерела загроз, уразливості, можливі наслідки тощо.

Один із прикладів списку найімовірніших загроз [13] містить такі загрози:

- хакерські атаки на центральний сервер;
- вплив вірусних і троянських програм на роботу системи;
- перехоплення інформації, що передається дротовими і бездротовими каналами зв'язку;
- доступ зловмисника з правами адміністратора на центральний сервер за допомогою розкрадання паролів та інших реквізитів розмежування доступу;
- доступ до мережі неавторизованих користувачів;
- наявність порушників серед обслуговуючого персоналу;
- помилки користувача;
- крадіжка (зловмисне виведення) з ладу апаратури;
- перебої в мережі електроживлення;
- стихійні лиха;
- поломка апаратури системи;
- помилки програмного забезпечення;
- витік інформації через побічні електромагнітні випромінювання і наведення;
- витік інформації акустоелектричним каналом.

1.5. Об'єкт і методи дослідження

Мета роботи - проєктування інформаційної системи моніторингу та оцінки загроз інформаційній безпеці (ІБ) технології "Smart Home" і розробка прототипу спроєктованої системи.

Досліджуваний об'єкт - системи "Smart Home" і загрози інформаційній безпеці, що виникають у них. Оскільки загрози інформаційній безпеці залежать від методів побудови системи, технологій, що використовуються, та інформаційних потоків, що обробляються, досліджувані системи SH було звужено до систем "Smart Home", що застосовуються в житлових будинках.

Проєктована система моніторингу та оцінювання загроз інформаційній безпеці технології "Smart Home" базується на моделі засобу інформаційної безпеки, виокремленій у процесі аналізу літературних джерел. Модель проєктованої системи:

- опис моделі системи SH;
- опис моделі загроз інформаційної безпеки;
- розроблення методів оцінювання загроз;
- розроблення автоматичного механізму моніторингу стану захисту SH.

Для проєктування системи інформаційної безпеки використовуються основні методи системного аналізу та технології розробки програмного забезпечення. Системний аналіз застосовують для розв'язання важко формалізованих і слабко структурованих проблем для зведення складної проблеми до взаємопов'язаної ієрархії простіших завдань [14].

Унаслідок аналітичного огляду предметної області було ухвалено рішення використовувати спосіб опису системи "Smart Home" шляхом поділу системи на підсистеми, оскільки цей спосіб використовують багато дослідників і виробників систем SH. У системі "Smart Home" було виділено такі підсистеми:

- підсистема управління та зв'язку;
- підсистема безпеки та моніторингу;
- підсистема електроживлення;
- підсистема освітлення;
- підсистема опалення;
- підсистема вентиляції;
- підсистема кондиціонування;
- підсистема мультимедіа.

Далі пропонується розбиття підсистем на компоненти, об'єкти управління та елементи: датчик і виконавчий механізм. Компоненти відповідають за роботу з об'єктами управління, а в об'єкта управління має бути вказано тип підсистеми. Усі об'єкти управління в SH мають датчик для зчитування необхідних даних і виконавчий механізм, що виконує будь-яку дію. Показник датчика і дія виконавчого механізму безпосередньо впливають на стан об'єкта і стан інформаційної безпеки всієї системи SH, тому для них мають визначатися обмеження.

Отримання даних про склад системи SH передбачається такими способами:

1. Користувач самостійно описує кожен елемент системи SH.
2. Система ІБ автоматично збирає дані з системи SH та визначає на їх основі склад системи SH.

Для побудови класифікації загроз інформаційній безпеці технології SH було вирішено використати список найімовірніших загроз, описаний у попередньому розділі, і використовувати такі критерії:

- величина відхилення значення датчика або виконавчого механізму;
- можливі джерела загрози;
- можливі наслідки загрози;
- уразливості.

Список загроз експерти визначають для різних об'єктів управління залежно від підсистем, у яких вони розташовуються, і залежно від ступеня відхилення даних датчика або виконавчого механізму.

Метод моніторингу стану SH ґрунтується на таких етапах:

- опис системи SH користувача;
- отримання даних із системи SH;
- аналіз даних, порівняння зі встановленими обмеженнями;
- визначення підсистеми та об'єкта, у якому виникла загроза;
- оцінка загрози за визначеним експертами списком загроз.

Для опису моделі системи SH і загроз у проєктованій системі обрано мову XML, для опису структури даних файлів обрано мову XML-Schema. Вибір мов опису даних зумовлений великою вкладеністю і великим обсягом даних, формат

XML ефективний у роботі саме з такими даними. XML - розширювана мова розмітки, яка дає змогу створити будь-яку необхідну для конкретної сфери застосування розмітку. Структуру XML-файлу можна описати за допомогою специфікації XML-Schema, яка визначає правила документа. XML має реалізації "парсерів" (програми для аналізу розмітки) для всіх сучасних мов програмування. Також XML підтримується на низькому апаратному, мікропрограмному та програмному рівнях у сучасних апаратних рішеннях.

Для розробки прототипу спроєктованої системи обрано:

– середовище розробки Microsoft Visual Studio 2023;

– мова програмування C#;

– система для побудови клієнтського додатка Windows Presentation Foundation (WPF).

Вибір інструментів зумовлений насамперед наявним досвідом роботи.

Вибір версії середовища розробки визначено виходячи з підтримки цієї версії новішими версіями, наявності при цьому сучасного базового функціоналу та широкого поширення. Додатковим плюсом середовищ розробки Visual Studio є наявність інструментів для роботи з XML і XML-Schema файлами, зокрема автоматична перевірка під час редагування XML-файлу, якщо в ньому вказано XML-Schema.

Систему WPF обрано через можливість використання будь-якого .NET-сумісної мови програмування разом із мовою XAML. WPF і XAML об'єднуються в повнофункціональну систему подання для створення візуально привабливих класичних додатків Windows, що включають в себе користувальницький інтерфейс, мультимедіа та складні бізнес-моделі.

Мова програмування C# має кілька класів для ефективної та швидкої роботи з XML-документами, які можна обрати залежно від поставлених завдань.

РОЗДІЛ 2
КЛАСИФІКАЦІЯ ЗАГРОЗ ІНФОРМАЦІЙНІЙ БЕЗПЕЦІ СИСТЕМИ "SMART HOME"

Для опису класифікації загроз інформаційній безпеці технології "Smart Home" використовують найвірогідніші загрози, уразливості, можливі наслідки [13] і критерії аналізу загроз [15].

Вибір списку найімовірніших загроз було визначено в попередньому розділі. Було виокремлено три критерії для аналізу загроз: джерело загрози, вразливості безпеки інформації, підсистема SH, у якій виникла загроза. Критерії джерело загрози та вразливості безпеки інформації поділяються на категорії та підкатегорії.

Категорії, підкатегорії та приклади:

1. Джерело загрози

a. Антропогенні джерела

i. Зовнішні (I.A.1 - I.A.6):

– кримінальні структури, потенційні злочинці, хакери та інші.

ii. Внутрішні (I.B.1 - I.B.4):

– основний персонал, допоміжний персонал та інші.

b. Техногенні джерела

i. Зовнішні (II.A.1 - II.A.3):

– засоби зв'язку, мережі інженерних комунікації та інші.

ii. Внутрішні (II.B.1 - II.B.4):

– неякісні технічні засоби обробки інформації, неякісні програмні засоби обробки інформації та інші.

c. Стихійні джерела (III.A.1 - III.A.7)

- пожежі, землетруси, повені, урагани та інші.

2. Уразливості безпеки інформації

a. Об'єктивні (A.I - A.IV):

- супутні технічним засобам випромінювання, що визначаються особливостями елементів, які визначаються особливостями об'єкта, що захищається, та інші.

b. Суб'єктивні (B.I, B.II):

- помилки та порушення.

c. Випадкові (C.I, C.II):

- збої, відмови та пошкодження.

Після визначення необхідних даних було побудовано класифікацію загроз інформаційній безпеці технології "Smart Home". Фрагмент отриманої класифікації представлено в таблиці 2.1.

Таблиця 2.1

Фрагмент класифікації загроз

Загроза (тип атаки)	Джерело загрози	Уразливість	Уразливості безпеки інформації	Можливі наслідки	Підсистема
Хакерські атаки на центральний сервер	I.A.1, I.A.2, I.A.3, I.A.4, I.A.5, I.A.6, I.B.1, I.B.2, I.B.3, I.B.4	Підключення мережі "Розумного будинку" до Інтернету. Відсутність (неефективність) механізмів захисту периметра мережі	A.I. a, A.I.b A.II. a, A.II.b, A.III.b, A.IV.b, B.I.a, B.I.b, B.I.c, B.II.a, B.II.c, B.II.d C.I.a, C.I.c	Порушення роботи або вихід з ладу центрального сервера і всієї системи. Порушення конфіденційності, цілісності та доступності інформації (КЦД).	управління та зв'язку
Перехоплення інформації, що передається дротовими і бездротовими каналами зв'язку	I.A.1, I.A.2, I.A.3, I.A.4, I.A.5, I.A.6, I.B.1, I.B.2, I.B.3, I.B.4 II.A.1, II.B.1, II.B.2	Можливість доступу зловмисника до дротових каналів або до зони стійкого перехоплення радіосигналів мережі. Відсутність (неефективність) механізмів захисту трафіку	A.I. a, A.I.b A.II. a, A.II.b A.III. a, A.III.b A.IV. a, A.IV.b B.I.a, B.I.b, B.I.c B.II.a, B.II.b B.II.c, B.II.d C.I.a,C.I.b,C.I.c, C.I.d, C.II.b	Порушення конфіденційності інформації, що передається каналом. Можливий доступ до управління системою.	управління та зв'язку; безпеки та моніторингу; електроживлення; освітлення; опалення; вентиляції; кондиціонування; мультимедіа;
Доступ зловмисника з правами адміністратора на центральний сервер за допомогою розкрадання паролів та інших реквізитів розмежування доступу	I.A.1, I.A.2, I.A.3, I.A.4, I.A.5, I.A.6, I.B.1, I.B.2, I.B.3, I.B.4	Відсутність (неефективність) механізмів аутентифікації та ідентифікації	A.I.a, A.I.b, A.II.a, A.II.b, A.III.a, A.III.b,A.IV.b B.I.a, B.I.b, B.I.c B.II.a, B.II.b, B.II.c, B.II.d, C.I.a,C.I.b,C.I.c, C.I.d,C.II.a,C.II.b	Порушення КЦД інформації, що знаходиться всередині мережі.	управління та зв'язку;

РОЗДІЛ 3
ПРОЄКТУВАННЯ СИСТЕМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

3.1. Проєктування системи інформаційної безпеки

Проєктування інформаційної системи полягає у визначенні функціональних вимог користувача системи, визначенні основних компонентів майбутньої системи та етапів її роботи, розробленні алгоритмів. Для розроблення описаних моделей використовується уніфікована мова моделювання UML.

Функціональні вимоги. Для опису функціональних вимог до розроблюваної системи було побудовано діаграму варіантів використання (рис. 3.1).

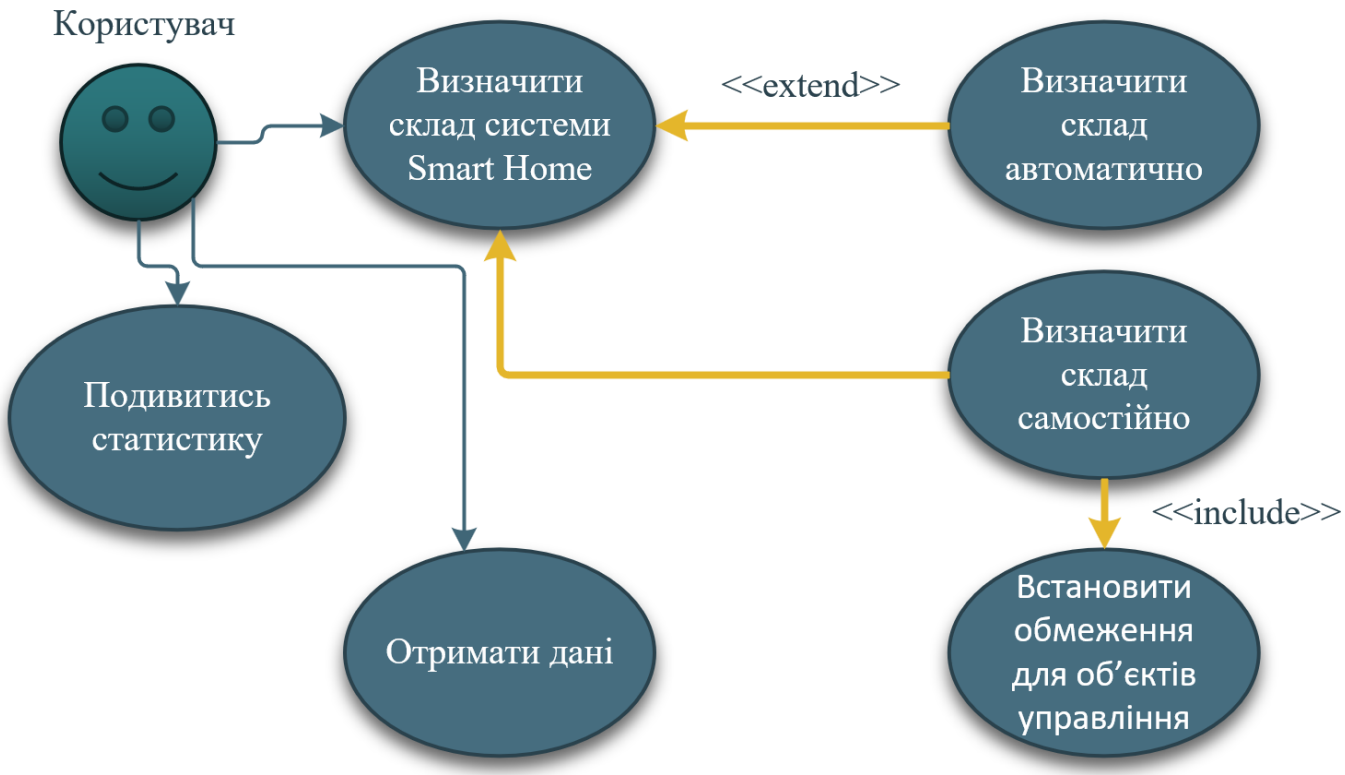

Рис. 3.1. Діаграма варіантів використання

Діаграма варіантів використання показує основні функціональні вимоги:

1. Визначити систему SH:

a. автоматично, шляхом визначення "ідеального" стану SH;

b. визначити самостійно, встановлюючи при цьому обмеження для об'єктів управління;

2. Отримати дані про склад SH;

3. Переглянути статистику про виявлені загрози;

3.2. Компоненти системи

На основі функціональних вимог було виділено основні етапи роботи проєктованої системи:

1. Визначення складу системи SH і встановлення обмежень одним зі способів:

a. отримання "ідеального" стану системи SH,

b. додавання користувачем вручну кожного елемента системи SH і обмежень для них.

2. Отримання даних із системи SH.

3. Перевірка всіх отриманих даних (моніторинг) у режимі реального часу.

4. Генерація сповіщень про стан системи SH.

На основі отриманих даних було визначено основні компоненти системи ІБ і побудовано діаграму компонентів (рис. 3.2).

Рис. 3.2. Діаграма компонентів

Докладніше про обрані компоненти системи:

- компонент визначення системи необхідний для визначення складу системи SH з отриманих даних;
- компонент представлення даних слугує для відображення користувачеві отриманих системою ІБ даних;
- компонент встановлення обмежень необхідний для автоматичного визначення обмежень для показань елементів системи ПД з отриманих даних;
- база даних слугує для зберігання даних про склад системи SH, класифікацію загроз ІБ і даних, що використовуються системою ІБ;
- компонент моніторингу здійснює моніторинг стану системи SH, шляхом перевірки вхідних даних і визначення невідповідностей;
- компонент сповіщень служить для інформування користувача про виявлені загрози або підозрілі дії.

3.3. Алгоритми системи

Після визначення функціональних вимог до системи інформаційної безпеки, основних етапів роботи системи та основних компонентів системи було розроблено алгоритми та побудовано діаграми діяльності для таких основних дій системи ІБ:

1. Алгоритм визначення користувачем складу системи SH і встановлення обмежень для об'єктів управління.
2. Алгоритм автоматичного визначення складу системи SH і обмежень об'єктів управління.
3. Алгоритм моніторингу стану системи SH.

Алгоритм визначення складу системи SH користувачем. Даний алгоритм виконується системою ІБ при визначенні складу системи SH і встановлення обмежень шляхом додавання користувачем вручну кожного елемента системи SH і обмежень для них.

Для відображення алгоритму було побудовано діаграму діяльності (рисунок 3.3.).

Алгоритм автоматичного визначення складу системи SH. Алгоритм виконується системою ІБ під час визначення складу системи SH та встановлення обмежень шляхом отримання "ідеального" стану системи SH. Для відображення алгоритму було побудовано діаграму діяльності (рис. 3.4).

Алгоритм моніторингу стану системи SH. Алгоритм полягає в перевірці всіх отриманих даних із системи SH у режимі реального часу. Для відображення алгоритму було побудовано діаграму діяльності (рис. 3.5).

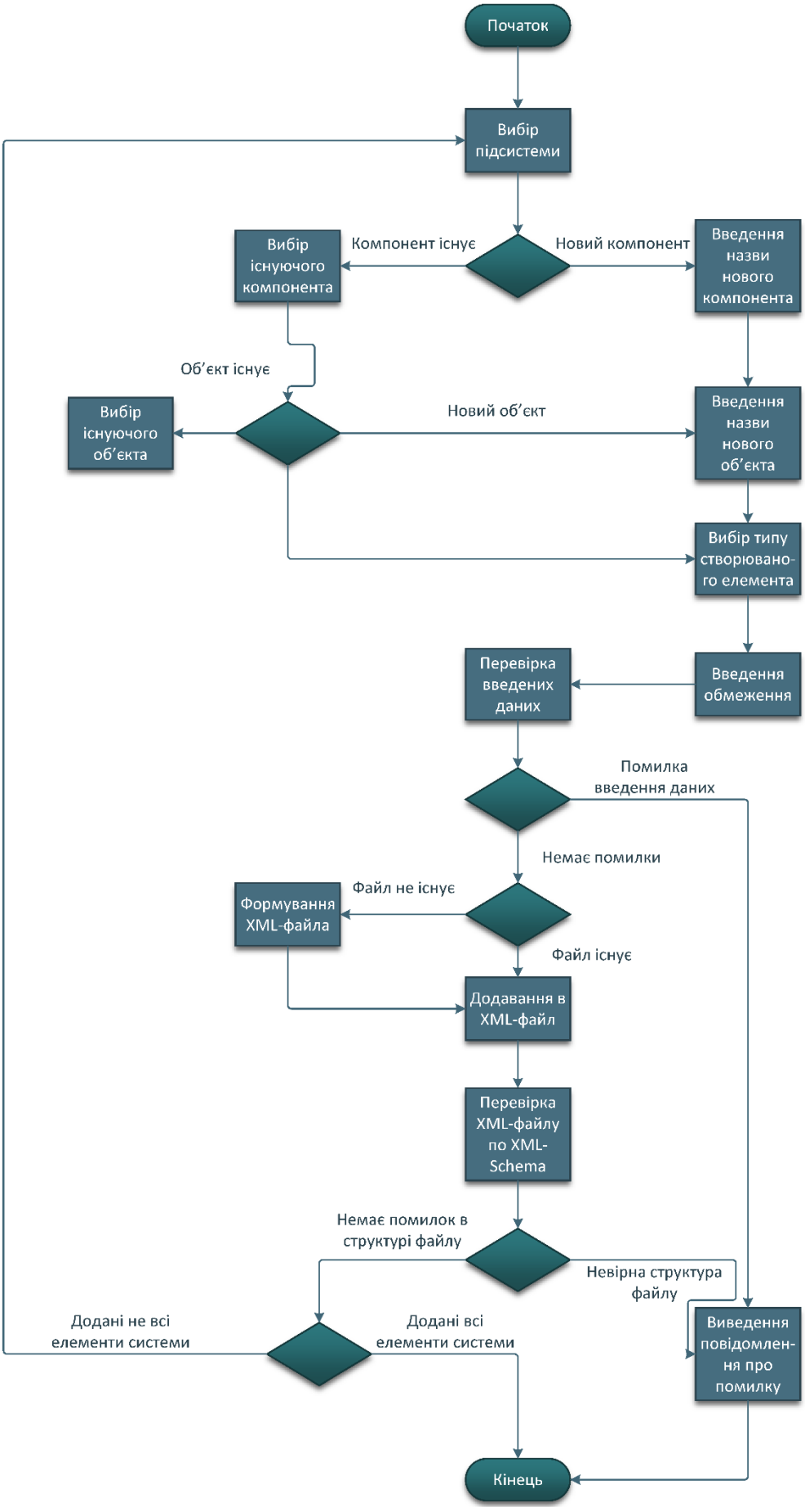

Рис. 3.3. Алгоритм визначення користувачем складу системи SH

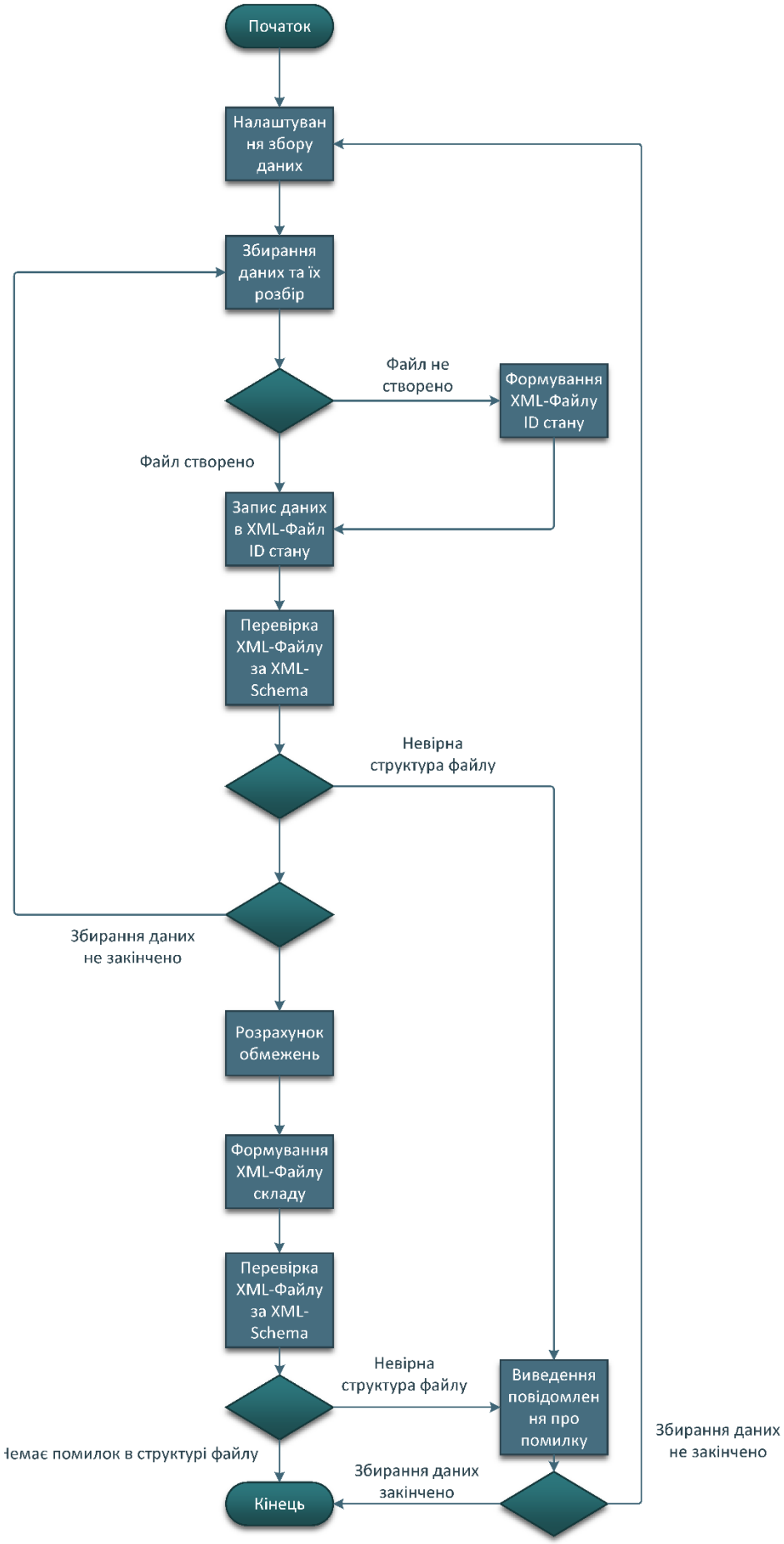

Рис. 3.4. Алгоритм автоматичного визначення складу системи ПД

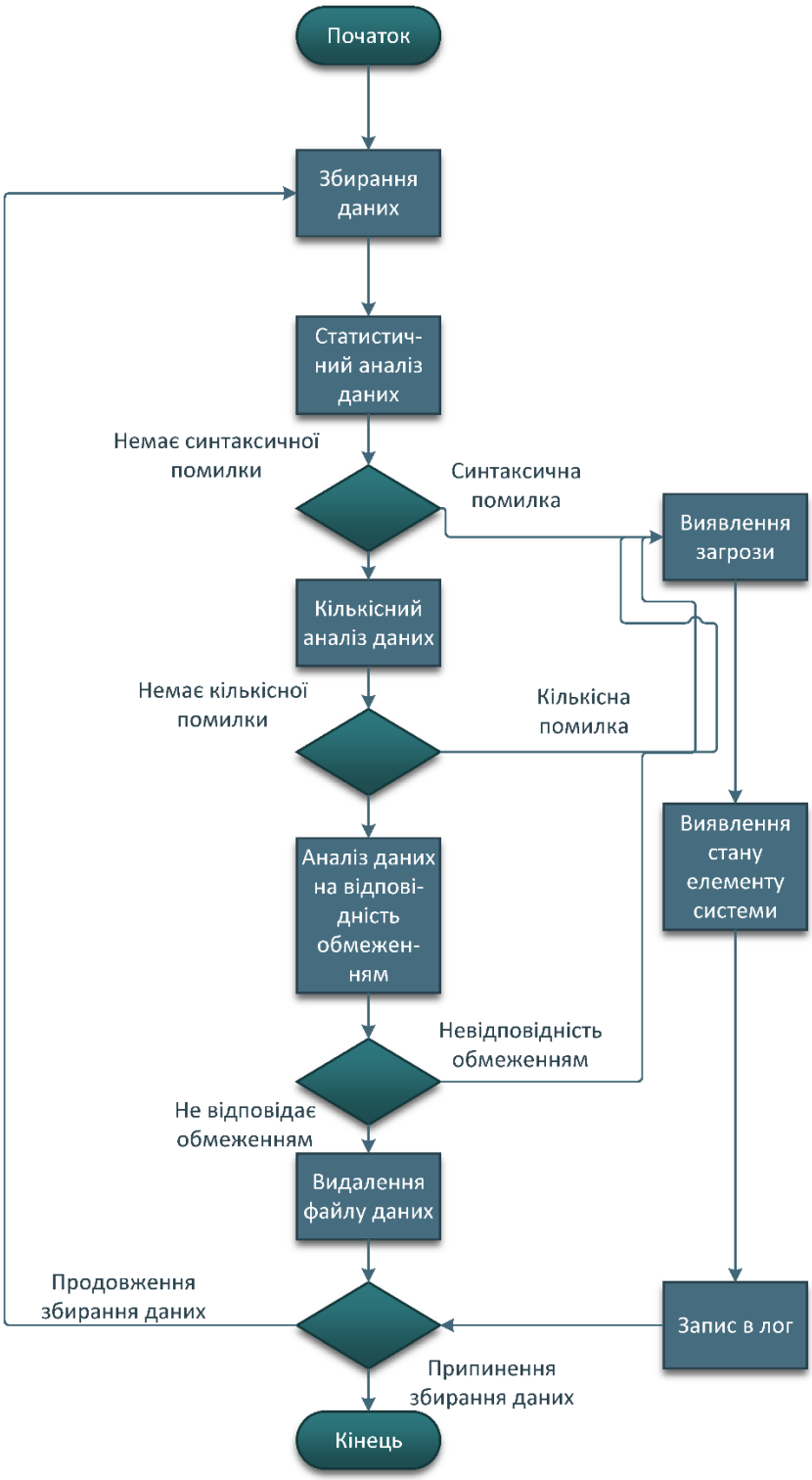

Рис. 3.5. Алгоритм моніторингу стану системи SH

3.4. Структура опису даних

У проєктованій системі інформаційної безпеки для опису даних про склад системи "Smart Home", обмеження показань елементів, що проєктуються системи і для

опису загроз інформаційної безпеки обрано формат XML. Структури XML-файлів було описано мовою XML Schema.

Структура даних для опису складу системи "Smart Home". Під час визначення складу системи SH користувачем самостійно, шляхом визначення кожного елемента системи, формується XML-файл опису системи SH. Графічне представлення схеми опису складу системи SH показано на рисунку 3.6, зміст файлу наведено в додатку А.

Використовувані позначення:

1. System - система SH;
2. Subsystem - підсистема;
3. Component - компонент підсистеми;
4. Object - об'єкт управління;
5. Sensor - датчик, Actuator - виконавчий механізм;
6. Name - назва, Type - тип елемента;
7. Min - мінімальне значення, Max - максимальне значення;
8. Average - обмеження.

Наступні елементи мають атрибути:

– System, атрибут Name (назва системи SH);

– Subsystem/Component/Object/Sensor/Actuator, атрибут ID (ідентифікатор);

– Average, атрибут Same (визначає, чи є обмеження постійним);

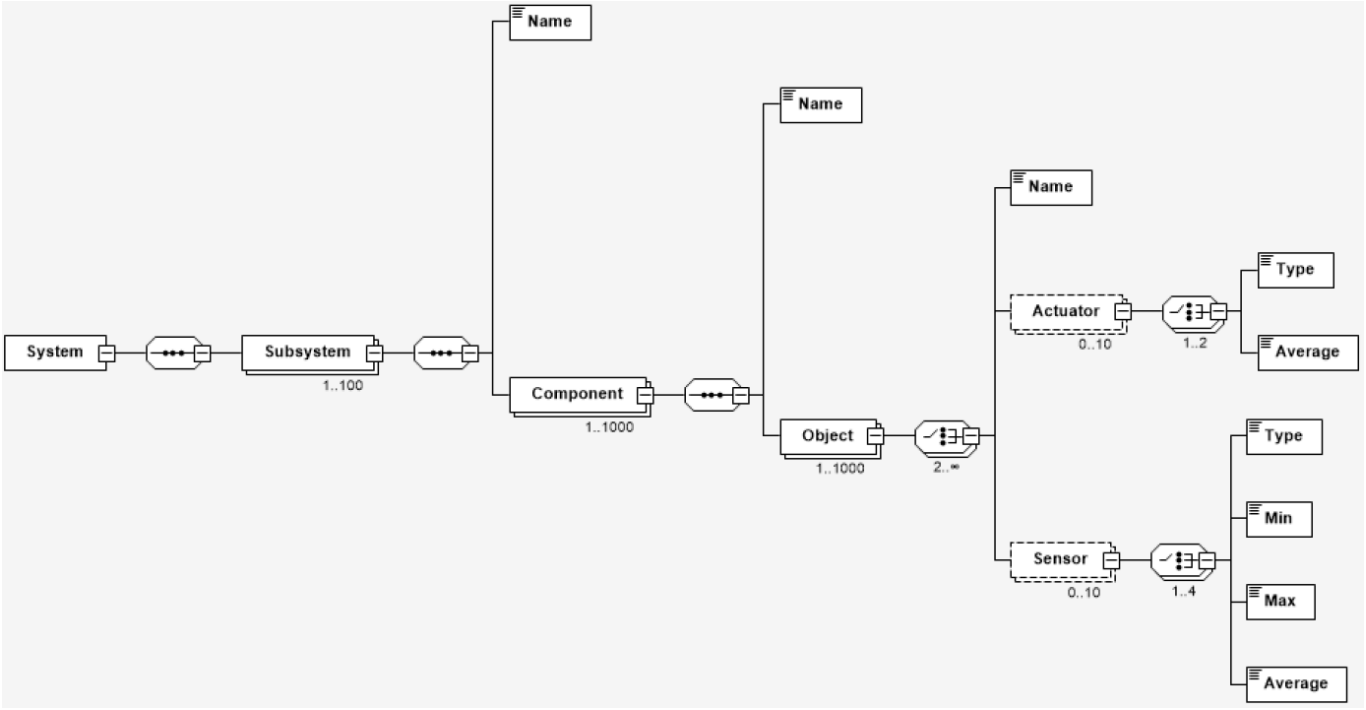

Рис. 3.6. Графічне представлення схеми опису складу системи "Smart Home"

Структура даних для опису "ідеального" стану системи "Smart Home". За автоматичного способу визначення складу системи SH спочатку формують файл з "ідеальним" станом системи, потім розраховують обмеження для елементів системи, і формують файл з описом складу системи SH. Графічне представлення файлу, що описує структуру "ідеального" стану системи SH представлено на рис. 3.7., зміст файлу наведено в додатку Б.

Використовувані позначення:

1. System - система SH;
2. Subsystem - підсистема;
3. Component - компонент підсистеми;
4. Object - об'єкт управління;
5. Sensor - датчик, Actuator - виконавчий механізм;
6. Name - назва, Type - тип елемента;
7. Min - мінімальне значення, Max - максимальне значення;
8. Average - обмеження,
9. Values/Value - значення/значення показника елемента.

Наступні елементи мають атрибути:

- Система, атрибути:
- Name (назва системи SH), DateFrom/DateTo (дата початку/дата закінчення збору даних), EverySec (інтервал збору даних), Round (ступінь округлення значення показника елемента);
- Subsystem/Component/Object/Sensor/Actuator, атрибут ID - ідентифікатор;
- Average, атрибут Same - визначає, чи є обмеження постійним;
- Value, атрибут DateTime - дата і час отримання даних;

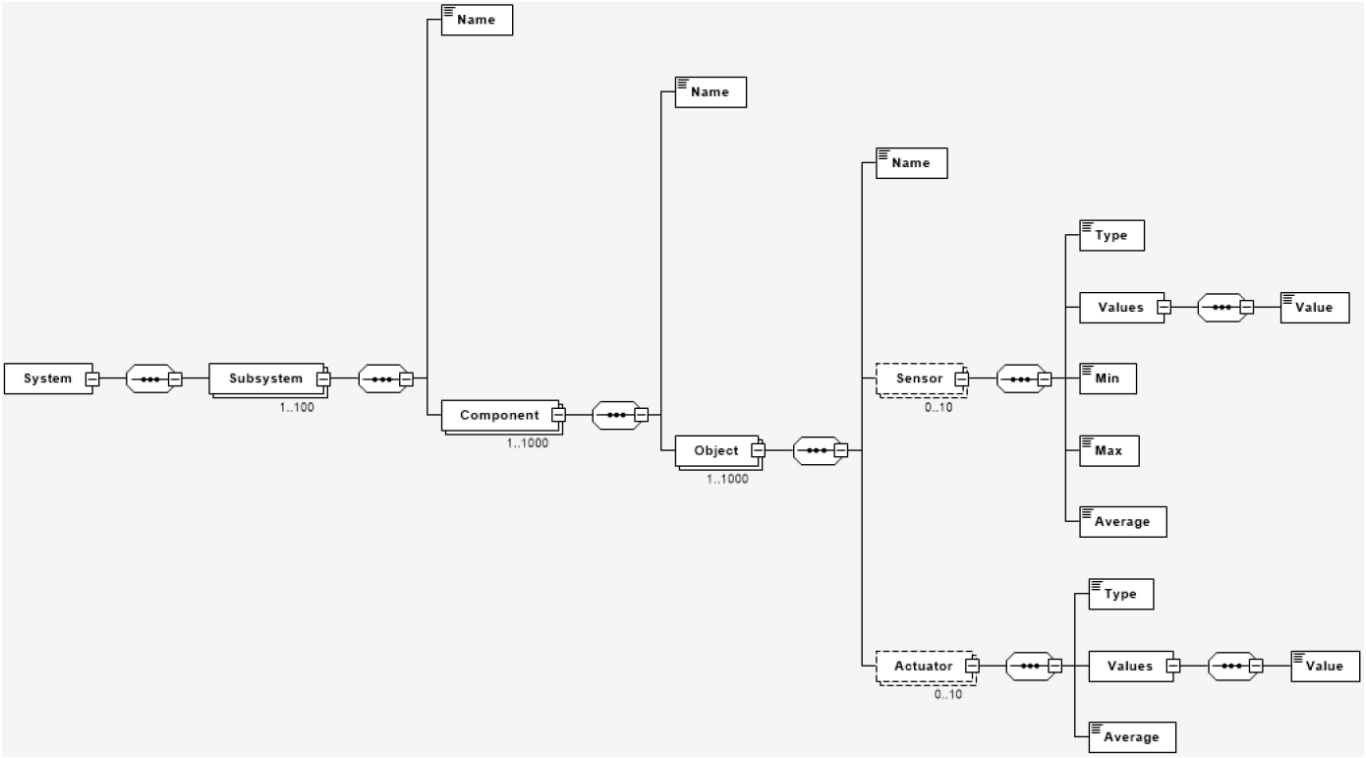

Рис. 3.7. Графічне представлення схеми опису "ідеального" стану системи SH

Структура даних для опису загроз системи "Smart Home". Графічне представлення схеми опису загроз ІБ системи SH показано на рис. 3.8., зміст файлу наведено в додатку В. Використовувані позначення:

1. Threats - загрози, Threat - загроза;
2. Object - об'єкт управління;
3. Sensor - датчик;
4. Actuator - виконавчий механізм;
5. Condition - умова;

6. Consequences - можливі наслідки, Consequence - можливий наслідок;
7. Sources - джерела, Source - джерело;
8. Causes - можливі причини, Cause - можлива причина.

Наступні елементи мають атрибути:

- Object, атрибути SubsystemID (ідентифікатор підсистеми) та
- Name (назва);
- Condition, атрибути Type (тип відхилення) та ID (ідентифікатор);
- Threat, атрибути Name (назва загрози), ID;
- Consequence/Source/Cause, атрибут ID.

Приклади типів відхилення:

- Тип = 00 (будь-яке відхилення від обмежень);
- Тип = 1 (0% - 20% відхилення від обмежень);
- Тип = 2 (20% - 40% відхилення від обмежень);
- Тип = 3 (40% - 60% відхилення від обмежень);
- Тип = 4 (60% - 80% відхилення від обмежень);
- Тип = 5 (80% - 100% відхилення від обмежень);
- Тип = 100 (100% відхилення від обмежень);

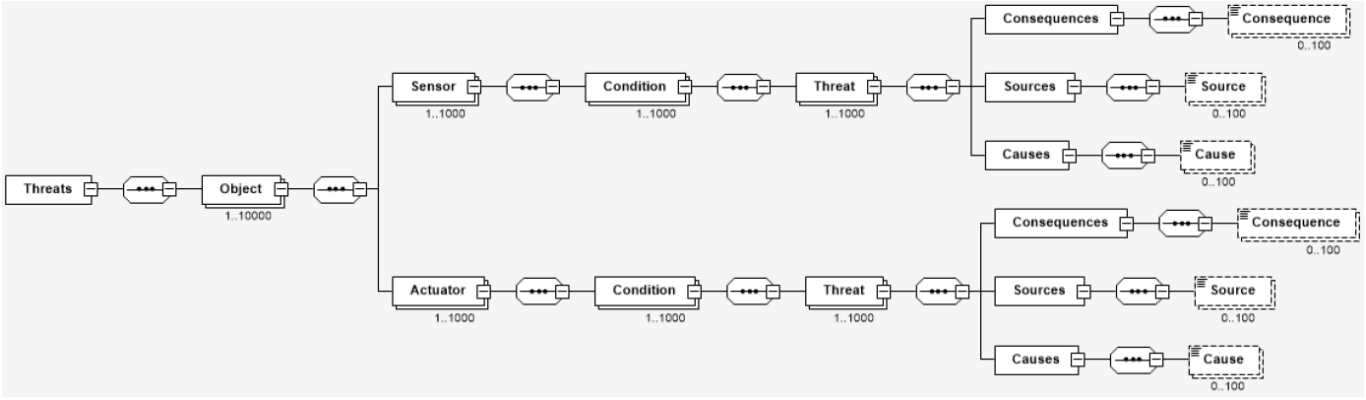

Рис. 3.8. Графічне представлення схеми опису загроз системи "Smart Home"

РОЗДІЛ 4
СИСТЕМА МОНІТОРИНГУ ТА ОЦІНКИ ЗАГРОЗ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ТЕХНОЛОГІЇ SMART HOME

4.1. Прототип системи інформаційної безпеки

У розроблюваному прототипі системи інформаційної безпеки для "розумного будинку" до описаних у попередньому розділі компонентів системи додано компонент генерування даних. Цей компонент використовує файл-зразок для визначення складу системи SH і значення показників об'єктів управління. Значення показників файлу-зразка використовуються під час генерації випадкових значень у файлах, що генеруються. Файли генеруються в окремому потоці програми. Рис. 4.1 демонструє потік генерування файлів (кожні дві секунди) з даними та запуску їх моніторингу.

```
private static void ThreadStartGenerateFiles()//потік для генерування файлів даних та
запуску їх моніторингу
        {
            while (do_generate)
            {
               mw.GenerateFile();//метод генерування файлів-даних

                if (do_monitoring)
                {
                   while (generated_files.Count != 0)//поки є файли для генерування
                    {
                        Monitoring();
                    }
                }
                Thread.Sleep(2000);//2sec
            }
                }
```

Рис. 4.1. Потік генерування та моніторингу файлів

Приклад вмісту файлу-зразка наведено на рис. 4.2.

```
<Date/Time/iD subsystem/iD component/iD object/A-actuator or s-sensor/iD actuator or sensor/type/value>
24-09-2023/00:00:00/4/1/1/A/1/s/1
24-09-2023/00:00:00/4/1/1/S/1/D/20
24-09-2023/00:00:00/4/1/2/S/2/D/30
24-09-2023/00:00:00/4/1/2/A/2/S/1
24-09-2023/00:00:OO/5/2/3/A/3/S/1
24-09-2023/00:00:OO/5/2/3/S/3/D/19
24-09-2023/00:00:00/7/3/4/S/4/D/18
24-09-2023/00:00:00/7/3/4/A/4/s/O
24-09-2023/00:00:00/6/4/5/S/5/D/0
24-09-2023/00:00:00/6/4/5/A/5/S/0
24-09-2023/00:00:OO/1/5/6/S/6/A/O
```

Рис. 4.2. Приклад файлу зразка

Під час запуску прототипу застосунку необхідно визначити склад системи SH, після чого стануть доступні кнопки для управління моніторингом. Інтерфейс основного вікна до визначення складу системи представлено на рис. 4.3.

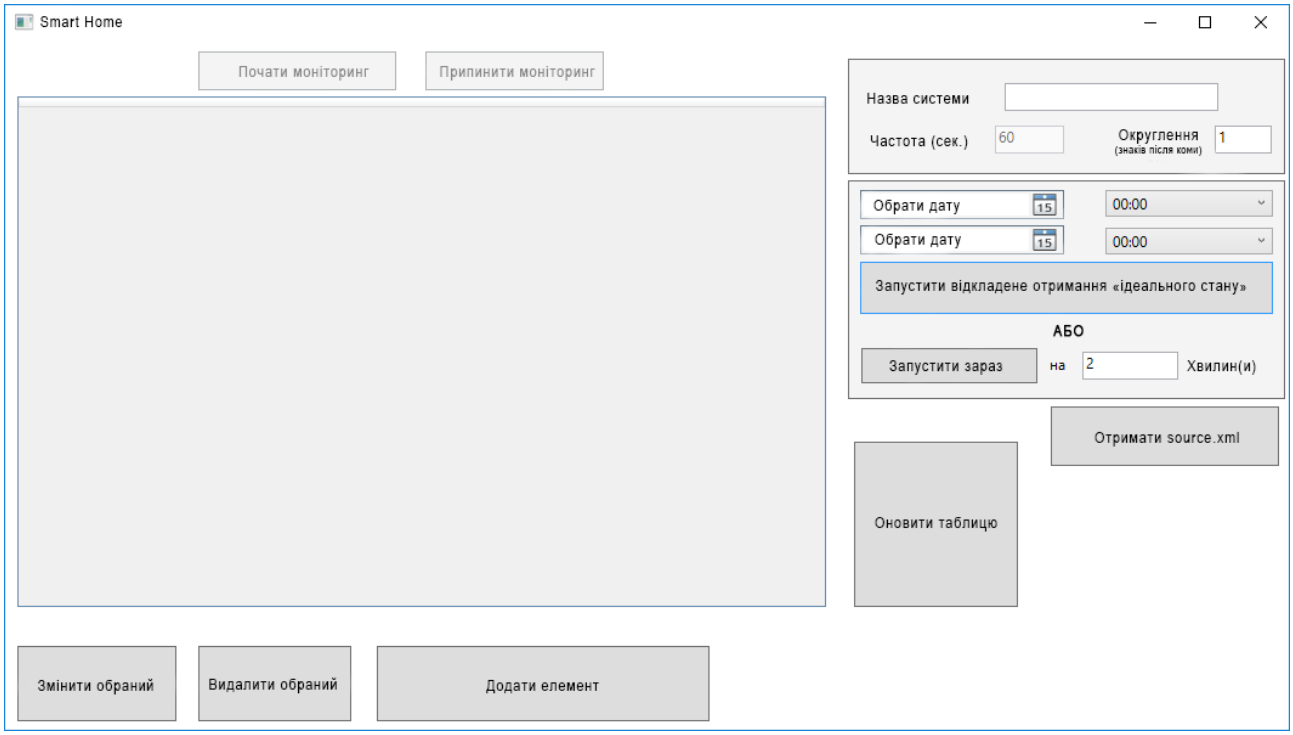

Рис. 4.3. Інтерфейс основного вікна до визначення складу системи

4.2. Формування "ідеального" стану системи "Smart Home"

Для автоматичного отримання складу системи SH необхідно в правій частині основного вікна визначити налаштування отримання "ідеального стану":

- вказати назву системи SH;

- визначити частоту збору даних (частота генерування файлів із даними);
- вказати ступінь округлення значень показань (за замовчуванням встановлено округлення до одного знака після коми);
- позначити дату і час початку та закінчення відкладеного збору даних або визначити час закінчення для збору даних у поточний момент часу.

Панель налаштування автоматичного збору даних представлена на рис. 4.4.

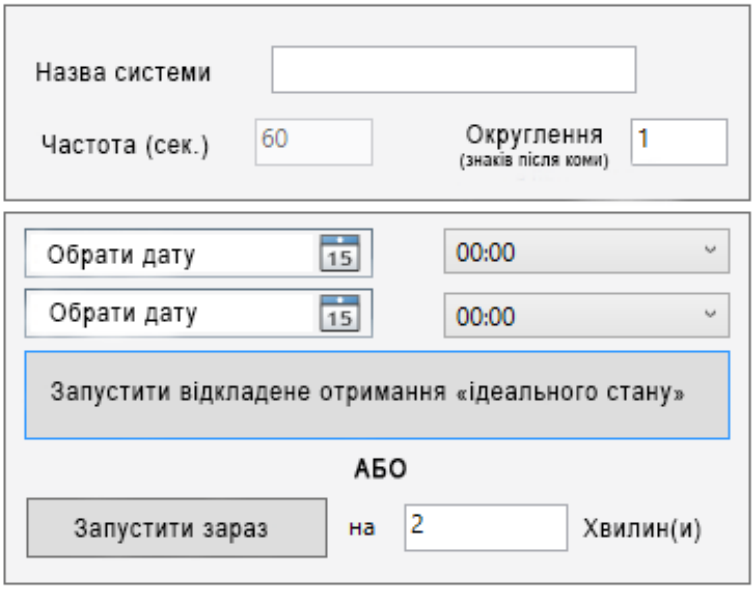

Рис. 4.4. Панель налаштування автоматичного збору даних

У додатку Г подано лістинг методу отримання "ідеального стану". Рис. 4.5. демонструє фрагмент коду, що додає елементи датчик або виконавчий механізм і дані елементів в XML-файл "ідеальний стан".

```
//додавання нового елементу з одним значенням
if (xdoc.Descendants(elem_tagname).Where(x => x.Attribute("ID").Value == elem_id).Count() ==
0)
{
        XmlElement elem = doc.CreateElement(elem_tagname);
        elem.SetAttribute("ID", elem_id);
        root_for_element.AppendChild(elem);

        XmlElement elem1 = doc.CreateElement("Type");
        string type_ = "";
        if (elem_tagname == "Actuator")//actuator
        {
                type_ = Actuator_types[type];
        }
        else if (elem_tagname == "Sensor")//sensor
        {
                type_ = Sensor_types[type];
        }
        elem1.InnerText = type_;
        elem.AppendChild(elem1);
        XmlElement elem2 = doc.CreateElement("Values");
        elem.AppendChild(elem2);
        element_for_value = elem2;
```

Рис. 4.5. Фрагмент методу додавання елементів

4.3. Визначення складу системи "Smart Home"

Для самостійного визначення користувачем складу системи SH необхідно на основному вікні вибрати кнопку "Додати елемент", після чого відкриється вікно для додавання/редагування елемента. Інтерфейс вікна додавання елемента представлений на рис. 4.6.

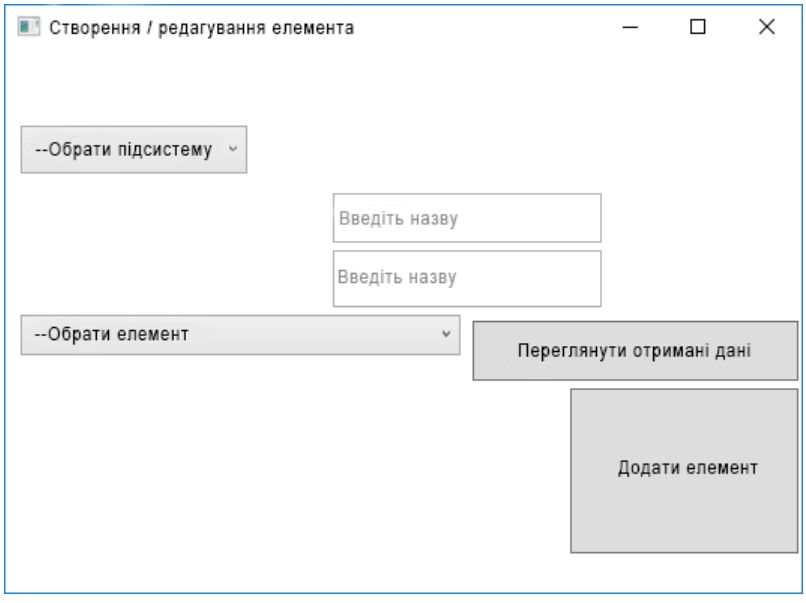

Рис. 4.6. Інтерфейс вікна додавання елемента

Користувачеві необхідно вибрати зі списку, що випадає, підсистему, ввести назви для компонента підсистеми та об'єкта управління (або вибрати з існуючих, якщо вони були додані раніше), вибрати створюваний елемент. Після вибору елемента стануть доступні список, що випадає, для визначення типу елемента, текстові поля для визначення обмежень. Приклад заповненої форми для створення датчика елемента управління лампочки представлено на рисунку 4.7. Лістинг методу для додавання нових елементів представлений у додатку Д.

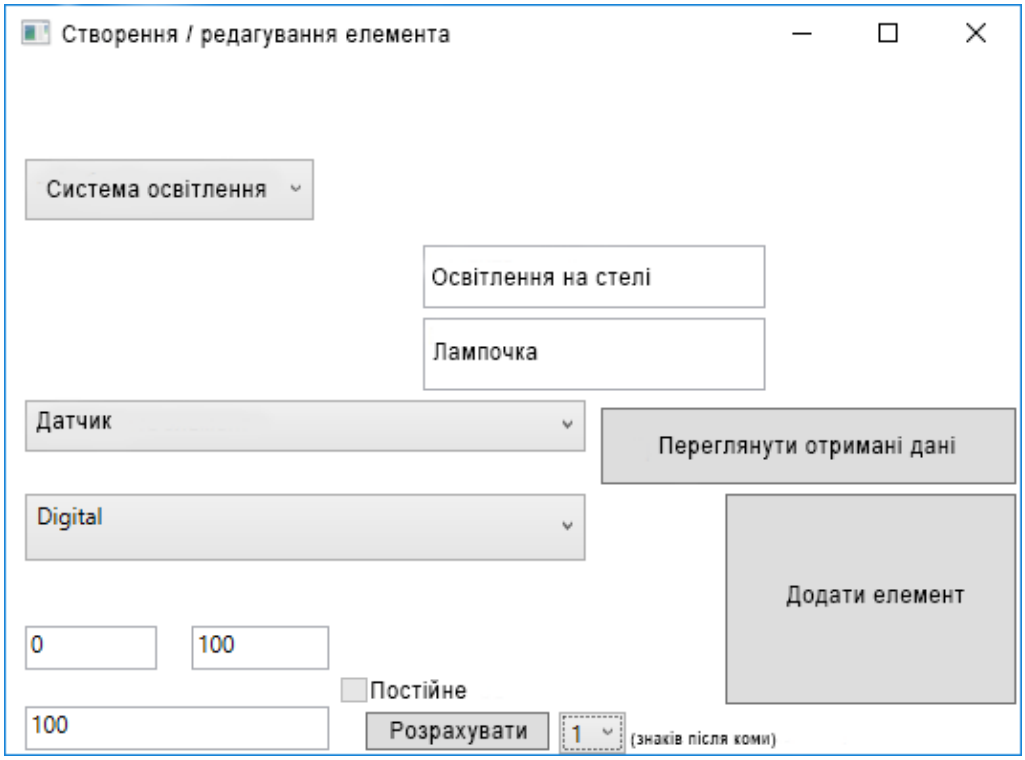

Рис. 4.7. Інтерфейс заповненого вікна додавання елемента

Після визначення складу системи в таблиці на основному вікні додатка з'являться дані про елементи. Дані в таблиці можна сортувати за спаданням і зростанням будь-якого стовпця. Приклад заповненої таблиці з даними наведено у таблиці 4.1.

Таблиця 4.1

Дані з сенсорів

SubsystemName	SubsystemID	ComponentName	ComponentID	ObjectName	ObjectID	Average	Min	Max	ID	Type	Name	State
Система освітлення	4	Освітлення на стелі	1	Лампочка	1	83	...	...	1	Switch	Actuator	Green
Система освітлення	4	Освітлення на стелі	1	Лампочка	1	24,5	11	38	1	Digital	Sensor	Green
Система освітлення	4	Освітлення на стелі	1	Лампочка	2	33,7	5	54	2	Digital	Sensor	Green
Система освітлення	4	Освітлення на стелі	1	Лампочка	2	83	...	...	2	Switch	Actuator	Green
Система опалення	5	Опалення кімнати	2	Батарея	3	16,2	7	28	3	Digital	Sensor	Green
Система опалення	5	Опалення кімнати	2	Батарея	4	83	...	...	3	Switch	Actuator	Green
Система кондиціювання	7	Загальне кондиціювання	3	Кондиціонер	5	18,2	1	30	4	Digital	Sensor	Green
Система кондиціювання	7	Загальне кондиціювання	3	Кондиціонер	5	16	...	...	4	Switch	Actuator	Green
Система вентиляції	6	Вентиляція вікон	4	Вікно	6	0	0	30	5	Digital	Sensor	Green
Система вентиляції	6	Вентиляція вікон	4	Вікно	6	16	...	...	5	Switch	Actuator	Green
Система керування і зв'язку	1	Сервер	5	Системний блок	6	1	0	1	6	Analog	Sensor	Green

Рисунок 4.8 демонструє метод для заповнення таблиці даними з XML-файлу зі складом системи.

```
private void fulltable()//заповнення таблиці даними з source.xml
        {
           //дані з xml
            XDocument doc = XDocument.Load(path);
            var result = doc.Descendants("Average").Select(x => new
            {
                SubsystemName = x.Parent.Parent.Parent.Parent.Element("Name").Value,
                SubsystemID = x.Parent.Parent.Parent.Parent.Attribute("ID").Value,
                ComponentName = x.Parent.Parent.Parent.Element("Name").Value,
                ComponentID = x.Parent.Parent.Parent.Attribute("ID").Value,
                ObjectName = x.Parent.Parent.Element("Name").Value,
                ObjectID = x.Parent.Parent.Attribute("ID").Value,
                Average = x.Value,
                Min = x.Parent.Element("Min") != null ? x.Parent.Element("Min").Value : "--
",
                Max = x.Parent.Element("Max") != null ? x.Parent.Element("Max").Value : "--
",
                ID = x.Parent.Attribute("ID").Value,
                Type = x.Parent.Element("Type").Value,
                Name = x.Parent.Name.ToString(),
                State = "green"
            });
           dgrid.ItemsSource = result;//заповнюємо таблицю

            elements_count = dgrid.Items.Count;//записуємо кількість елементів таблиці РД
           FullLists();//заповнюємо списки з наявним ID підсистем,компонентів,об'єктів,
елементів
                }
```

Рис. 4.8. Метод для заповнення таблиці

Такі дані елементів, як назва та обмеження, можна змінювати, для цього необхідно виконати подвійне натискання на необхідній комірці або вибрати рядок у таблиці та натиснути кнопку "Змінити вибраний". У першому варіанті відкриється вікно

для редагування конкретного значення даних елемента (рис. 4.9), у другому відкриється заповнене даними елемента вікно додавання/редагування елемента, описане раніше.

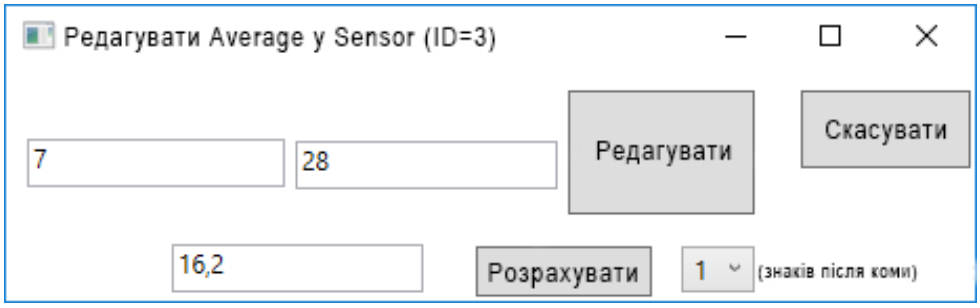

Рис. 4.9. Приклад редагування комірки таблиці

4.4. Моніторинг стану системи "Smart Home"

Для запуску моніторингу стану системи SH на головному вікні необхідно вибрати кнопку "Запустити моніторинг". За необхідності моніторинг стану SH можна залишити, вибравши кнопку "Зупинити моніторинг". У разі виявлення загроз і невідповідності показань датчика або виконавчого механізму об'єктів керування, у таблиці з даними змінюється значення елемента в стовпчику "State" (статус) і рядок елемента підсвічується кольором. Приклад зовнішнього вигляду таблиці з даними, у яких виявлено загрози, наведено в таблиці 4.2.

Таблиця 4.2

Приклад таблиці з виявленими загрозами

SubsystemName	SubsystemID	ComponentName	ComponentID	ObjectName	ObjectID	Average	Min	Max	ID	Type	Name	State
Система освітлення	4	Освітлення на стелі	1	Лампочка	1	83	...	...	1	Switch	Actuator	Green
Система освітлення	4	Освітлення на стелі	1	Лампочка	1	24,5	11	38	1	Digital	Sensor	Red
Система освітлення	4	Освітлення на стелі	1	Лампочка	2	33,7	5	54	2	Digital	Sensor	Red
Система освітлення	4	Освітлення на стелі	1	Лампочка	2	83	...	...	2	Switch	Actuator	Green
Система опалення	5	Опалення кімнати	2	Батарея	3	16,2	7	28	3	Digital	Sensor	Red
Система опалення	5	Опалення кімнати	2	Батарея	4	83	...	...	3	Switch	Actuator	Green
Система кондиціювання	7	Загальне кондиціювання	3	Кондиціонер	5	18,2	1	30	4	Digital	Sensor	Red
Система кондиціювання	7	Загальне кондиціювання	3	Кондиціонер	5	16	...	...	4	Switch	Actuator	Green
Система вентиляції	6	Вентиляція вікон	4	Вікно	6	0	0	30	5	Digital	Sensor	Red
Система вентиляції	6	Вентиляція вікон	4	Вікно	6	16	...	...	5	Switch	Actuator	Green
Система керування і зв'язку	1	Сервер	5	Системний блок	6	1	0	1	6	Analog	Sensor	Red

Лістинг методу для моніторингу даних наведено в додатку Е.

Метод для визначення загрози представлено на рис. 4.10. У методі перевірки даних у разі виявлення помилок визначається їхній вид, список описаних помилок показано в таблиці 4.3.

```
private static int GetThreatID(string tagname, string elemid, int dev)//визначення ID
загрози
        {
            int threat_id = -1;
            XDocument xdoc = XDocument.Load(path);
            string name="";

            switch (tagname)
            {
                case ("S"):
                    tagname = "Sensor";
                    break;
                case ("A"):
                    tagname = "Actuator";
                    break;
            }
           //отримуємо назву об'єкта, що становить загрозу
```

Рис. 4.10. Метод визначення загрози

Таблиця 4.3

Список помилок

№ помилки	Вид помилки
1	Помилка у форматі дати
2	Помилка у форматі часу
3	Помилка у форматі ID підсистеми
4	Помилка у форматі ID компонента
5	Помилка у форматі ID об'єкта
6	Помилка у форматі виду елемента
7	Помилка у форматі ID елемента
8	Помилка у форматі типу елемента
9	Помилка у форматі значення елемента
33	Не знайдена підсистема із заданим ID
44	Не знайдено компонент із заданим ID
55	Не знайдено об'єкт із заданим ID
771	Не знайдено датчик із заданим ID
772	Не знайдено виконавчий механізм із заданим ID
8810	У датчика із заданим ID не вказано тип
881	Неправильний тип датчика
8820	У виконавчого механізму із заданим ID не вказано тип
882	Неправильний тип виконавчого механізму
99	Дані сенсора не відповідають обмеженням
100	Неправильна кількість елементів даних
103	У вхідному масиві відсутні дані по підсистемі
104	У вхідному масиві відсутні дані щодо компонента
105	У вхідному масиві відсутні дані щодо об'єкта
1071	У вхідному масиві відсутні дані за датчиком
1072	У вхідному масиві відсутні дані по виконавчому механізму

Подвійне натискання на статус елемента відкриває вікно з докладними відомостями про можливу загрозу.

За відсутності даних про загрозу система ІБ представляє користувачеві дані про помилку і дату її виявлення.

РОЗДІЛ 5
ОХОРОНА ПРАЦІ

5.1. Аналіз небезпечних і шкідливих факторів, що впливають на інженера

Відділ проектування знаходиться на другому поверсі п’ятиповерхового будинку. Приміщення має розміри: довжина 8 м, ширина 4 м, висота 4. Загальна площа - 32 м^2, загальний об’єм – 128 м^3. У відділі знаходиться 5 робочих місць інженерів-проектувальників, оснащені комп’ютерами.

Робоча площа одного співробітника становить:

$$S_{\text{роб}} = \frac{S_{\text{заг.пл}}}{N} = \frac{32}{5} = 6{,}4 \text{ м}^2$$

Робочий об’єм одного співробітника:

$$V_{\text{роб}} = \frac{V_{\text{заг.об}}}{N} = \frac{128}{5} = 25{,}6 \text{ м}^3$$

N- кількість співробітників у відділі

$S_{\text{заг.пл}}$ – загальна площа;

$V_{\text{заг.об}}$ – загальний об’єм.

Відповідно до [44] площа на одне робоче місце має становити не менше ніж 6 м^2, а об'єм не менше ніж 20 м^3. Робоче місце інженера-проектувальника відповідає вимогам.

В проектному відділі інженера-проектувальника знаходяться: комп’ютери, принтер. У даному приміщенні температура повітря у теплий період року становить 30°С, використовується природне та штучне освітлення. Штучне освітлення виконано у вигляді перериввчастих ліній світлодіодних світильників. Рівень шуму в приміщенні становить 54 дБ, а згідно з Державними санітарними нормами [45] не повинен перевищувати 50 дБ.

Робоче місце розташоване так, щоб природне світло падало з лівої сторони, при цьому відстань зі світлом до робочого місця - 1 м. Висота робочої поверхні столу над підлогою 750 мм, глибина столу – 800 мм, ширина столу 1300мм. Робочий стіл має простір для ніг висотою 650 мм та шириною 600 мм.

Перелік шкідливих та небезпечних виробничих чинників.

Створення сприятливих умов праці, в роботі інженера-проектувальника, має велике значення як для полегшення , так і для підвищення продуктивності праці. Відповідно до [46] шкідливими виробничими факторами є:

1. Підвищена температура робочого приміщення
2. Недостатня освітленість робочої поверхні
3. Виробничий шум
4. Електромагнітні випромінювання радіочастотного діапазону
5. Іонізуючі випромінювання

Відповідно до [47] робота інженера-проектувальника у приміщенні з енерговитратами 90-120 ккал/год. відносяться до категорії легких фізичних робіт Іа (роботи, що виконуються сидячи і не потребують фізичного напруження).

Таблиця 5.1

Оптимальні величини температури

Період року	Категорія робіт	Температура повітря,℃
Холодний період року	Легка Іа	22-24
Теплий період року		23-25

Допустимі величини температури на постійних робочих місцях:

Період року	Категорія робіт	Температура повітря,℃	
		Верхня межа	Нижня межа
Холодний період року	Легка Іа	25	21
Теплий період року		28	22

У проектному відділі температура повітря становить 30℃ в теплий період року,що перевищує допустиму на 2 ℃. Забезпечили температуру приміщення 23 ℃,за допомогою механічної вентиляції з вентилятором VORTICE VARIO, повітрообмін якого становить 680 $м^3$ /год.

Недостатня освітленість. В приміщенні встановлені персональні комп’ютери, присутнє природне та штучне освітлення . За вимогами [48], величина коефіцієнта природної освітленості повинна бути не менше 1.5%. В проектному відділі порушенні

вимоги, освітленість робочої поверхні складає 370 лк , а коефіцієнт освітленості складає 1.2%. Природне світло проникає у приміщення через бічні світло прорізи. Вікна мають жалюзі. Штучне освітлення виконано у вигляді переривчастих ліній світлодіодних світильників, розташованих паралельно лінії зору інженера-проектувальника. Для місцевого освітлення використовувати галогенні лампи розжарювання

Виробничий шум. Шум на робочому місці створюється: комп'ютером та периферійним пристроєм. Допустимі рівні звукового тиску на робочому місці повинні відповідати вимогам [49]:

Таблиця 5.2

Санітарні норми виробничого шуму, ультразвуку та інфразвуку

Вид трудової діяльності, робоче місце	Рівні шуму та еквівалентні рівні шуму, ДБА, дБАекв
Конструювання та проектування.	50

Реальний рівень шуму в проектному відділі становить 54 дБ, що перевищує допустимий рівень.

Для зменшення рівня шуму рекомендується використовувати місцеву та загальну звукоізоляцію, шумопоглинаючі екрани, поглинаючі фільтри.

5.2. Організаційні та конструктивно-технологічні заходи для зниження впливу шкідливих виробничих факторів

Нормалізація повітря робочої зони. Для створення й автоматичної підтримки в ІТ відділі незалежно від зовнішніх умов оптимальних значень температури, вологості, чистоти і швидкості руху повітря, у холодний час року використовується водяне опалення, у теплий час року застосовується кондиціонування повітря [50].

Виробниче освітлення. Під час аналізу освітлення на робочому місті програміста було встановлено, що воно не відповідає встановленим нормам, тому для покращення умов праці рекомендуємо збільшити рівень загальної освітленості приміщення

шляхом встановлення 5 додаткових світильників, щоб загальна кількість лам відповідала розрахованому вище значенню, а саме 36 світлодіодних ламп. Також для підтримки запроектованого освітлення у чистому виді необхідно скласти графік, де передбачити очищення віконних блоків і світильників не менше 2 разів на рік [51].

Електробезпека. Електробезпечність у приміщенні ІТ відділу пропоную забезпечити наступними технічними способами і засобами захисту:

- для зменшення накопичення статичної електрики застосовувати зволожувачі і нейтралізатори, антистатичне покриття підлоги;
- забезпечити приєднання металевих корпусів устаткування до жили, що заземлює. Заземлення корпуса ПК забезпечити підведенням жили, що заземлює, до розеток.

Опір заземлення 4 Ом, згідно для електроустановок з напругою до 1000 В. А також організаційними заходами:

- своєчасне проведення інструктажів з техніки безпеки [52].

Ергономіка та організація робочого місця. Після проведення аналізу робочого місця програміста в ІТ Відділі було з'ясовано, що воно відповідає встановленим вимогам.

Виходячи з результатів аналізу важкості та напруженості праці пропоную скоротити час роботи за комп'ютером, робити перерви сумарний час яких повинен складати 50 хвилин при 8-ми годинному робочому дні [53].

5.2.1. Розрахунок повітрообміну за надлишком тепла у проектному відділі

Приміщення має розміри 4×8×4, яке розміщується на другому поверсі п’ятиповерхового будинку з південного боку. Площа вікон F = 2,88 м^2. На вікнах розміщенні жалюзі. У приміщенні 5 інженерів-проектувальників, розташовано $N_{пк} = 5$ персональних комп’ютерів та принтер. Для штучного освітлення використовується 4 офісних світлодіодних світильника потужністю 125 Вт.

1. Розраховуємо загальну кількість тепла:

$$Q_{над} = Q_{осв} + Q_{облад} + Q_{ін\text{-}пр.} + Q_{рад,}\ \text{Вт} \qquad (5.1)$$

$Q_{над}$ – загальна кількість тепла

$Q_{осв}$ - кількість тепла від джерел штучного освітлення

$Q_{облад}$ - кількість тепла від обладнання

$Q_{ін\text{-}пр.}$- кількість тепла від інженерів-проектувальників

$Q_{рад,}$- кількість тепла від сонячної радіації

2. Розраховуємо кількість тепла від джерел штучного освітлення:

$$Q_{осв} = N \cdot \eta \,, \tag{5.2}$$

де N - сумарна потужність джерел освітлення, Вт; η - коефіцієнт теплових витрат (η = 0,55 – для світлодіодних ламп).

$$Q_{осв.} = 125 \cdot 4 \cdot 0{,}55 = 275 \text{ Вт}$$

2. Розраховуємо кількість тепла при роботі обладнання: 5 комп'ютерів і принтера (в режимі друку):

$$Q_{облад} = n \cdot P_{комп.} + P_{пр.}, \tag{5.3}$$

де n – кількість комп'ютерів (обладнання);

$P_{комп}$ – встановлена потужність комп'ютерів, $P_{комп} = 400$ Вт

$P_{пр.}$ – потужність принтера в режимі друку, $P_{пр} = 465$ Вт

$$Q_{облад} = 5 \cdot 400 + 465 = 2.5 \text{ кВт}$$

3. Розраховуємо кількість тепла від інженерів-проектувальників:

$$Q_{ін\text{-}пр.} = n \cdot q \text{ , Вт} \tag{5.4}$$

n – кількість інженерів-проектувальників

q – кількість тепла, що виділяється одним інженером-проектувальником

Кількість тепла, що виділяється одним інженером-проектувальником, який виконує легку фізичну роботу дорівнює 99 Вт.

$$Q_{ін\text{-}пр} = 5 \cdot 99 = 495 \text{ Вт}$$

4. Розраховуємо кількість тепла від сонячної радіації:

$$Q_{рад} = m \cdot S \cdot k \cdot q_{скл} \tag{5.5}$$

де m – число вікон; $S_{вікна}$ – площа одного вікна, $S_{вікна} = 2{,}88\ м^2$;

k – коефіцієнт, віконного переплетення: $k = 0{,}6$ матові;

$q_{скл.}$ – надходження тепла через $1\ м^2$ вікна при різній орієнтації вікон: $q_{скл.} = 150$ – південь;

$$Q_{рад}=1\cdot2,88\cdot0,6\cdot150=259,2 \text{ Вт}$$

5. Загальна кількість тепла в проектному відділі:

$$Q_{над}= Q_{осв}+ Q_{облад}+ Q_{ін\text{-}пр.} + Q_{рад}=275+2500+495+259,2 = 3,529 \text{ кВт}$$

6. Потрібний повітрообмін за надлишком тепла:

$$L = \frac{Q}{c\cdot\rho\cdot(t_{вид}-t_{зовн})}, \text{м}^3/\text{год} \quad (5.6)$$

Q - кількість тепла, яке виділяється в приміщення за годину, Дж:

$$Q=3600\cdot Q_{надл}=3600\cdot3529=12704 \text{ Вт} = 5328 \text{ кДж};$$

c – теплоємність повітря, Дж/кг (в інтервалі температур від 0ºС до 100ºС приймається рівною $1,01\cdot10^3$Дж/кг);

ρ – густина повітря, кг/м^3(дорівнює $\rho_{внт}$=1,2 кг/м^3);

$t_{вид}$– температура повітря, що видаляється, $t_{вид}=30$℃

$t_{зовн.}$- температура повітря, що подається до робочої зони, $t_{зовн.}$=23℃

$$L = \frac{5328}{1,01\cdot10^3\cdot1,2\cdot(30-23)} = 628 \text{ м}^3/\text{год}$$

Оскільки, в проектному відділі підвищена температура повітря на 2 ℃ від допустимого значення 28℃, встановили механічну вентиляцію з вентилятором VORTICE VARIO , яка забезпечила надходження до приміщення температури повітря 23 ℃, дане значення є оптимальним.

5.3. Пожежна безпека

Відповідно до [54] дане приміщення відноситься до категорії В по вибухово-пожежній та пожежній небезпеці із-за використання у ньому твердих горючих матеріалів з температурою спалаху понад 61℃.

Проектний відділ оснащено:

- Двома безпровідними датчиками детектування диму SD-02 (оповіщає при задимленні приміщення; площа обслуговування: до 20 м^2);

- двома порошковими вогнегасниками ВП-5 (для приміщення категорії В за відсутності горючих газів і рідин, площею до 50 м^2 і масою вогнегасної речовини – 5 кг, мінімальна кількість порошкових вогнегасників 2).
- LifeSOS LS-30LR бездротова пожежно-охоронна система (при детектуванні вторгнення, датчики передають на центральний блок сигнал тривоги по радіоканалу без проводів. Централь приймає сигнал від датчиків, включає сирену, відправляє інформацію на пульт централізованого нагляду, дзвонить на зазначені телефонні номери та відправляє SMS повідомлення з повідомленнями про тривогу.)

Для попередження виникнення пожеж проводяться організаційно-технічні заходи пожежної безпеки, які включають:

- включення питань пожежної безпеки у всі інструкції по техніці безпеки;
- виконання встановленого режиму експлуатації електричних мереж та обладнання;
- заборона куріння в недозволеному місці;
- видання необхідних інструктажів, планів евакуації.

План евакуації складається з графічної і текстової частин. Графічна частина являє собою схематичний план поверху (рис. 5.1), в якому зеленими суцільними стрілками вказують шляхи евакуації, що ведуть до основних евакуаційних виходів, а пунктирними зеленими стрілками - до аварійних виходів. Двері на шляху евакуації відчиняються назовні у напрямку виходу з будівлі. На плані евакуації умовними знаками показано розміщення вогнегасників, пожежних гідрантів, телефонів, аптечок медичної допомоги, електрощитів, датчиків диму, системи охоронно-пожежної сигналізації.

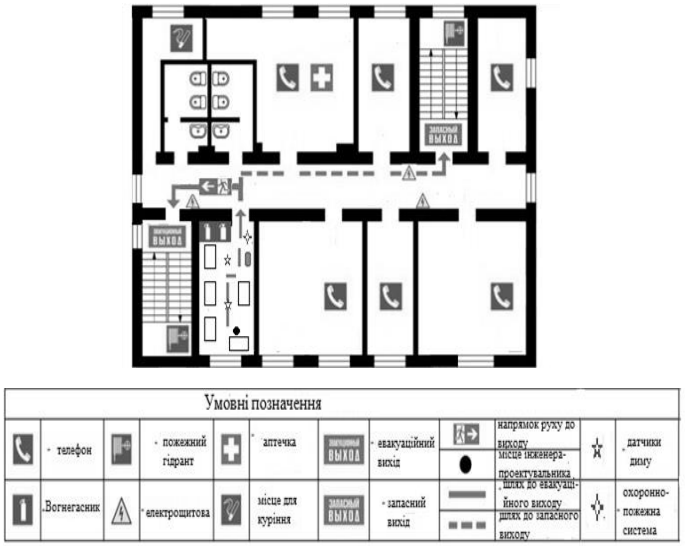

Рис 5.1. План евакуації 2 поверх

5.4. Інструкція з охорони праці при роботі з персональним комп'ютером

Вимоги безпеки перед початком роботи.

- Перед початком роботи працівник повинен зовнішнім оглядом перевірити цілісність корпусів системного блоку, відео монітора, принтера, клавіатури.
- Перевірити цілісність кабелів живлення, місць їх підключення (розеток електромережі, продовжувачів електромережі, розгалужувальних коробок, штепсельних вилок).
- Підготувати своє робоче місце, прибравши речі, які можуть заважати при виконанні роботи.
- Ввімкнути живлення ПК.
- У випадку, якщо після ввімкнення ПК не проходить загрузка або комп'ютер не виходить на робочий режим, працівник повинен повідомити керівника чи спеціаліста відділу інформаційних технологій.
- При виявленні ушкодження або яких-небудь інших недоліків повідомити безпосереднього керівника. Не приступати до роботи без його вказівки.
- Вимоги безпеки під час роботи
- Необхідно стійко розташувати всі складові пристрої на столі, в тому числі і клавіатуру. Разом з тим повинна бути передбачена можливість переміщення клавіатури.

ЇЇ розташування і кут нахилу повинні відповідати побажанням користувача ПК. Якщо в конструкції клавіатури не передбачений простір для опору долонь, то її слід розташовувати на відстані не менше 100 мм від краю столу в оптимальній зоні моніторного поля. При роботі на клавіатурі слід сидіти прямо, не напружуватись.

- Для зменшення несприятливого впливу на користувача пристроїв типу "миша" (вимушена поза, необхідність постійного контролю за якістю дій) слід забезпечити вільною більшу площу поверхні столу для переміщення "миші" і зручного упору ліктьового суглоба.
- Не припустимі сторонні розмови, роздратовуючи шуми тощо.
- Періодично при вимкнутому ПК слід видаляти злегка зволоженою мильним розчином хлопко-паперовою салфеткою пил з поверхонь апаратури. Екран і захисний екран протирають ватою, зволоженою спиртом.
- Не дозволяється використовувати рідинні або аерозольні засоби чистки поверхонь ПК.

Забороняється:

- самостійно ремонтувати апаратуру, в яких кінескоп та інші елементи можуть знаходитись під високою напругою (до 25 кВ0.)
- класти будь-які речі на апаратуру ПК, бутерброди та напої на клавіатуру або поруч з нею. Це може вивести її з ладу;
- затуляти вентиляційні отвори в апаратурі, це може призвести до її перегріву і виходу з ладу.

Для зменшення негативного впливу на стан здоров’я працівників різних факторів ризику, пов’язаних з роботою на ПК, передбачаються додаткові регламентовані перерви для відпочинку користувачів ПК:

- через кожний час безперервної роботи – 10 хвилин;
- через кожні 2 години – 15 хвилин.
- При можливості слід чергувати зміну діяльності з іншою, не пов’язаною з роботою на ПК.

- З метою зменшення негативного впливу монотонності доцільно застосовувати чергування операцій введення тексту і введення даних (зміна змісту і темпу роботи) і т.п.
- При роботі на лазерних принтерах:
- Розташовувати принтер необхідно поряд з системним блоком так, щоб з'єднувальні шнури не були натягнуті. Забороняється ставити принтер на системний блок.
- Перш, ніж програмувати роботу принтера, впевніться, що він знаходиться в режимі зв'язку з системним блоком.
- Для досягнення високоякісного, чистого, з високою роздільною здатністю зображення щоб не зіпсувати апарат, потрібно використовувати папір, марка якого вказана в інструкції до принтера (найчастіше папір вагою 60-135 г/м^2, типу Canon або Xerox 4024).
- Обрізання країв паперу повинно бути виконаним гострим лезом ножа, без заусенців – це зменшить вірогідність загинання паперу.
- При виконанні роботи (більше 20 хвилин), коли втручання користувача в роботу програми не потрібне, бажано вимикати живлення відео монітора.
- Для підтримки загального тонусу м'язів, профілактики кістково-м'язових порушень, зорового дискомфорту та інших несприятливих суб'єктивних почуттів під час регламентованих перерв необхідно виконувати комплекси рекомендованих вправ для очей, для хребта, для рук.
- Кількість мікро пауз до 1-2 хвилин слід визначити індивідуально. Форма та зміст перерв можуть бути різними виконання допоміжних робіт, не пов'язаних з роботою ПК , приймання їжі, виконання рекомендованих вправ.
- Виконання фізичних вправ протягом дня рекомендується індивідуально, залежно від почуття втоми. Гімнастика повинна біти на корекцію вимушеної пози покращення кровообігу, часткову компенсацію, дефіциту рухової активності.
- Про виявлені несправності (іскріння, пробоїв, запаху гару, ознак горіння тощо) негайно припинити роботу, відключити все обладнання від електромережі і терміново повідомити безпосереднього керівника або спеціаліста по ремонту ПК.
- Вимоги безпеки при закінчення роботи на ПК.

- Закінчити і зберегти в пам’яті ПК файли, які знаходились у роботі. Виконати всі дії для коректного завершення роботи в оперативній системі.
- Вимкнути принтер та інші периферійні пристрої, вимкнути системний блок. При наявності пристрою безперебійного живлення (ПБЖ) вимкнути його живлення.
- Вимкнути ПК кнопкою «POWER” (ЖИВЛЕННЯ) та вийняти штепсельну вилку кабелю живлення з розетки
- Накрити клавіатуру кришкою для попередження попадання в неї пилу.
- Навести порядок на робочому місці.

Вимоги безпеки в аварійних ситуаціях.

- Якщо після ввімкнення ПК відчувається запах горілого або при доторканні до металевих частин ПК відчувається дія електричного струму, потрібно негайно відключити ПК від електромережі та повідомити про це своєму керівникові.
- У випадку виникнення пожежі негайно розпочати гасіння наявними засобами пожежогасіння і повідомити за телефоном 101 (міська пожежна охорона) та начальнику ДПД підприємства. Пам`ятайте, що загашувати електроустановки слід вуглекислотними вогнегасниками, сухим піском, щоб уникнути ураження електричним струмом.

У разі виникнення інших аварійних ситуацій слід припинити роботу і повідомити про це керівника робіт.

ВИСНОВКИ ДО РОЗДІЛУ 5

На підставі виконаного розрахунку повітрообміну за надлишком тепла, значення якого 628 м^3/год, встановили механічну вентиляцію з вентилятором VORTICE VARIO, оскільки використання природної вентиляції є малоефективним. Механічна вентиляція здатна забезпечити виведення з проектного відділу температури 30℃ і підтримувати температуру повітря допустимого та навіть оптимального значення.

РОЗДІЛ 6
ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Охорона навколишнього середовища включає регулювання відносин у сфері збереження, використання та відтворення природних ресурсів, гарантування екологічної безпеки, уникнення та ліквідацію негативного впливу господарської та іншої діяльності на природне середовище, збереження природних ресурсів, генетичного фонду живої природи, ландшафтів та інших природних комплексів, унікальних територій та природних об'єктів, пов'язаних з історико-культурною спадщиною.

Використання сучасних програмно-апаратних платформ, що розробляються фахівцями з використанням комп’ютерної техніки, має бути супроводжене уважною увагою до охорони та збереження навколишнього середовища. Забруднення навколишнього середовища включає в себе дії, які вносять в екологічну систему живі або неживі компоненти, фізичні або структурні зміни, які порушують процеси круговороту і обміну речовин, а також відтоки енергії, що призводить до зниження продуктивності або руйнування даної екосистеми.

Забруднюючі речовини можна класифікувати за природою:

- фізичні забруднення, до яких відносяться: шумове забруднення і низькочастотна вібрація, електромагнітне забруднення, радіоактивні елементи;
- хімічні та біологічні забруднювачі, до яких відносяться: синтетичні органічні речовини, важкі метали, фтористі з’єднання;
- механічні, до яких відносяться: пил та тверді частки.

Для вирішення питань у сфері охорони навколишнього середовища важлива роль належить науці - екології. Екологія - це дослідження всіх взаємовідносин живих і неорганічних компонентів середовища та ефективного використання природних ресурсів. Основні завдання екології включають в себе повноцінну діагностику стану природи, прогнозування змін у природному середовищі, а також розробку профілактичних заходів з охорони навколишнього середовища.

У цьому розділі будуть обговорені аспекти забруднення навколишнього середовища, зокрема вплив інформаційних технологій та персональних комп'ютерів, а також розроблені ефективні методи та заходи щодо раціональної утилізації відпрацьованого обладнання без шкоди для природи.

6.1. Забруднення навколишнього середовища

Забруднення навколишнього середовища полягає у зміні характеристик навколишнього середовища (хімічних, механічних, фізичних, біологічних та пов'язаних з ними інформаційних), які виникають внаслідок природних або антропогенних процесів і призводять до погіршення функцій середовища щодо будь-якого біологічного чи технологічного об'єкта. Людина, використовуючи різні компоненти навколишнього середовища у своїй діяльності, впливає на його якість. Часто ці зміни виявляються у небажаній формі забруднення.

Сучасні технології, особливо інформаційні та телекомунікаційні, що інтегрують екологічні аспекти у своїй концепції, стали фундаментом інформаційного суспільства, перетворившись на основний спосіб життя для людства та ключовий елемент нового циклу розвитку цивілізації та планети.

Сучасні інформаційні технології є більш екологічно збалансованими порівняно з багатьма іншими аспектами людської діяльності. Проте їх ще нельзя повністю вважати екологічно безпечними. Наприклад, ефективність інформаційних мереж відразу залежить від кількості користувачів, тобто кількості підключених комп'ютерів. Але для виробництва одного звичайного персонального комп'ютера потрібно від 15 до 19 тонн матеріалів, що порівнюється з 25 тоннами, потрібними для виготовлення автомобіля. На кожен працюючий комп'ютер (використовуваний у середньому протягом 4 років) припадає 1,5 комп'ютери, які були вироблені. Близько третини комп'ютерів ніколи не буває продано взагалі через швидкість, з якою вони втрачають технологічну актуальність. Це означає, що затрачені ресурси справді наближаються до рівня автомобіля.

Електронні пристрої містять небезпечні токсичні сполуки, які, потрапляючи в навколишнє середовище, створюють серйозні загрози для життя людей. Наприклад, 22% ртуті, що видобувається щороку в усьому світі, використовується у електронній промисловості та міститься у мобільних телефонах. Кадмій, який є канцерогеном, використовується практично у всіх напівпровідникових пристроях. Свинець, особливо токсичний для нервової системи, міститься у акумуляторах та екранах моніторів. У міру розкладання захисних покриттів з електронних пристроїв у навколишнє середовище виділяються діоксин та інші високотоксичні сполуки.

Підвищена увага суспільства до екологічних проблем, а також більш суворі екологічні закони, змушують виробників обладнання створювати мережі для збору виведених з обігу технологічних засобів та підприємства з їх утилізації. Крім того, конструкція обладнання стає більш орієнтованою на можливість подальшої переробки матеріалів. Розмір мережі для утилізації "електронного сміття" залежить від регіону та місцевого законодавства.

Все оргтехнічне устаткування містить органічні компоненти (різні види пластику, матеріали на основі полівінілхлориду, фенолформальдегід) та повний спектр металів.

Отже, звичайний комп'ютер містить як цінні метали, такі як золото, срібло, алюміній, мідь, так і небезпечні, такі як кадмій, свинець, цинк, нікель. Тому при списанні та утилізації обладнання необхідно дотримуватися відповідних законодавчих вимог щодо охорони навколишнього середовища.

6.2. Заходи щодо запобігання забруднення навколишнього середовища

Персональні комп'ютери, ноутбуки та інша інформаційна техніка широко використовуються в наукових дослідженнях, промисловості та повсякденному домашньому користуванні. Проте з розвитком технологій ця техніка швидко застаріває, і на заміну приходять нові, потужніші та сучасніші моделі. Це призводить до зростання обсягів застарілої техніки, яка нагромаджується в підсобних приміщеннях та складах.

Утилізація оргтехніки та комп'ютерів включає кілька етапів. Перший етап - це списання обладнання безпосередньо на підприємстві. Далі відбувається розбір та сортування техніки для подальшого використання матеріалів. Деякі деталі можуть бути використані як вихідна сировина для вторинної переробки, зокрема, деякі елементи, що містять дорогоцінні метали.

Утилізація друкованих плат та електронних компонентів є важливим аспектом утилізації комп'ютерної техніки. Спеціальні методи, наприклад, використання спеціального розчину, дозволяють виділяти компоненти та використовувати їх повторно в новій продукції.

Проблема утилізації використаної комп'ютерної техніки набуває все більшого значення внаслідок зростання виробництва та частоти заміни техніки. Подальший прогрес у цій галузі може сприяти зменшенню податкового навантаження на компанії за утилізацію застарілої техніки, зробивши екологічну утилізацію економічно вигідною та сприятливою для довкілля.

ВИСНОВКИ ДО РОЗДІЛУ 6

Важливо бути пильними щодо охорони та збереження навколишнього середовища при використанні сучасних програмно-апаратних платформ, розроблених фахівцями з використанням комп’ютерної техніки. Розвиток та удосконалення інформаційних технологій повинні спрямовуватися не лише на створення максимально комфортних умов для людини, але й на досягнення безвідходного процесу утилізації відпрацьованої техніки, уникаючи негативного впливу на навколишнє середовище.

ВИСНОВКИ

Кваліфікаційна робота "Система моніторингу та оцінки загроз інформаційної безпеки технології Smart Home" висвітлила актуальну проблему забезпечення безпеки в галузі "розумних будинків". Швидкий технологічний прогрес та зростання популярності технологій Smart Home призвели до загострення кіберзагроз, що потребує системного підходу до забезпечення інформаційної безпеки.

Проведений аналіз технології "Smart Home" та наявних рішень захисту інформації систем " Smart Home " засвідчив відсутність єдиної методології опису систем SH, а отже, відсутність єдиної методології виявлення та оцінювання загроз інформаційній безпеці SH. У результаті аналізу було побудовано модель системи інформаційної безпеки для систем SH і сформовано список найімовірніших загроз інформаційній безпеці систем SH.

У ході роботи були проведені дослідження та аналіз існуючих систем безпеки Smart Home, виявлені основні загрози та вразливості, що можуть стати об'єктом кібератак. Використання комплексу методів, включаючи літературний аналіз, аналіз систем безпеки, емпіричні дослідження, спостереження та анкетування, експертні оцінки, моделювання та аналіз загроз та статистичний аналіз, дозволило отримати глибоке розуміння проблеми.

Запропоновано класифікацію ймовірних загроз інформаційній безпеці систем SH, яка пов'язує можливі загрози з об'єктами управління системи "Smart Home". Ця класифікація дає змогу визначати й оцінювати знайдені в системі SH загрози ІБ. Для збільшення кількості відомих загроз і поліпшення якості оцінювання загроз класифікація може бути доповнена експертами.

Спроєктовано систему інформаційної безпеки технології SH, розроблено алгоритми роботи системи, структури опису даних і прототип інформаційної системи. Основні функції системи: визначення складу системи SH і встановлення обмежень для показань об'єктів керування користувачем або автоматично, моніторинг даних системи SH і генерація сповіщень про стан системи SH. У разі існування в системі SH

невідомої загрози система сповіщає користувача про підозрілі дані. Розроблену систему використано для проведення імітаційних експериментів, під час яких у системі ІБ генерувалися вихідні дані для імітації дій системи SH, та аналізу виявлення можливих загроз ІБ.

В результаті досліджень була розроблена комп'ютерна програма, спрямована на моніторинг та оцінку загроз інформаційної безпеки технології Smart Home. Ця програма дозволяє реагувати на потенційні загрози та вчасно вживати відповідних заходів для запобігання атакам та захисту особистих даних користувачів.

Практичне значення отриманих результатів полягає в забезпеченні безпеки користувачів технології Smart Home, у збереженні конфіденційності їхніх особистих даних та у підвищенні довіри до цих технологій. Розроблена програма може бути використана для покращення існуючих систем безпеки та розробки нових, більш захищених рішень. Результати роботи доводять працездатність інформаційної системи і дають змогу зробити висновок про можливість розроблення засобу захисту систем "Smart Home", який користувач налаштовує під свою систему SH. Цей засіб захисту здійснює моніторинг стану всієї системи та оцінює знайдені загрози.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. J. Kumar and P. R. Ramesh, "Low Cost Energy Efficient Smart Security System with Information Stamping for IoT Networks," 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-5.
2. A. Roukounaki, S. Efremidis, J. Soldatos, J. Neises, T. Walloschke and N. Kefalakis, "Scalable and Configurable End-to-End Collection and Analysis of IoT Security Data : Towards End-to-End Security in IoT Systems," 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 2019, pp. 1-6.
3. M. S. Sharbaf, "IoT Driving New Business Model, and IoT Security, Privacy, and Awareness Challenges," 2022 IEEE 8th World Forum on Internet of Things (WF-IoT), Yokohama, Japan, 2022, pp. 1-4.
4. H. Garg and M. Dave, "Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-6.
5. E. P. Yadav, E. A. Mittal and H. Yadav, "IoT: Challenges and Issues in Indian Perspective," 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-5.
6. W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf and Y. A. Bangash, "An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10250-10276, Oct. 2020.
7. J. Singh, G. Singh and S. Negi, "Evaluating Security Principals and Technologies to Overcome Security Threats in IoT World," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 1405-1410.

8. A. K. Gupta and R. Johari, "IOT based Electrical Device Surveillance and Control System," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-5.
9. C. Sharma and N. K. Gondhi, "Communication Protocol Stack for Constrained IoT Systems," 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-6.
10. F. T. Jaigirdar, C. Rudolph and C. Bain, "Prov-IoT: A Security-Aware IoT Provenance Model," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 2020, pp. 1360-1367.
11. S. B. Sarvaiya and D. N. Satange, "Security in IP-Based IoT Node and Device Authentication," 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), Pune, India, 2022, pp. 1-5.
12. Y. Zheng, A. Pal, S. Abuadbba, S. R. Pokhrel, S. Nepal and H. Janicke, "Towards IoT Security Automation and Orchestration," 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Atlanta, GA, USA, 2020, pp. 55-63.
13. E. G. Maria Verzegnassi, K. Tountas, D. A. Pados and F. Cuomo, "Data Conformity Evaluation: A Novel Approach for IoT Security," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 842-846.
14. N. M. Karie, N. M. Sahri, W. Yang, C. Valli and V. R. Kebande, "A Review of Security Standards and Frameworks for IoT-Based Smart Environments," in IEEE Access, vol. 9, pp. 121975-121995, 2021.
15. D. Soni, V. Sharma and D. Srivastava, "Optimization of security issues in adoption of cloud ecosystem," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-5.

16. J. -Y. Yu and Y. -G. Kim, "Analysis of IoT Platform Security: A Survey," 2019 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea (South), 2019, pp. 1-5.
17. S. Ul Rehman, P. Singh, S. Manickam and S. Praptodiyono, "Towards Sustainable IoT Ecosystem," 2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE), Lombok, Indonesia, 2020, pp. 135-138.
18. Tina, Sonam, Harshit and M. Singla, "Smart Lightning and Security System," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-6.
19. A. Assiri and H. Almagwashi, "IoT Security and Privacy Issues," 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2018, pp. 1-5.
20. K. Abdul Sattar and A. Al-Omary, "A survey: security issues in IoT environment and IoT architecture," 3rd Smart Cities Symposium (SCS 2020), Online Conference, 2020, pp. 96-102.
21. J. S. Chavis, A. Buczak, A. Kunz, A. Rubin and L. Watkins, "A Capability for Autonomous IoT System Security: Pushing IoT Assurance to the Edge," 2020 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2020, pp. 256-261.
22. N. Rastogi, S. K. Singh and P. K. Singh, "Privacy and Security issues in Big Data: Through Indian Prospective," 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-11.
23. M. A. López Peña and I. Muñoz Fernández, "SAT-IoT: An Architectural Model for a High-Performance Fog/Edge/Cloud IoT Platform," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 633-638.
24. S. Shin and Y. Seto, "Development of IoT Security Exercise Contents for Cyber Security Exercise System," 2020 13th International Conference on Human System Interaction (HSI), Tokyo, Japan, 2020, pp. 1-6.

25. J. Fan, Y. Xu and J. Ma, "Research on Security Classification and Classification Method of Power Grid Data," 2022 6th International Conference on Smart Grid and Smart Cities (ICSGSC), Chengdu, China, 2022, pp. 72-76.
26. M. Y. Jamro, "IoT Security with QoS: Game changer for Industry and STEM Education," 2021 International Carnahan Conference on Security Technology (ICCST), Hatfield, United Kingdom, 2021, pp. 1-4.
27. M. A. El. zuway and H. M. Farkash, "Internet of Things Security: Requirements, Attacks on SH-IoT Platform," 2022 IEEE 21st international Ccnference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 2022, pp. 742-747.
28. A. M. Awadelkarim Mohamed and Y. Abdallah M. Hamad, "IoT Security: Review and Future Directions for Protection Models," 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 2020, pp. 1-4.
29. D. Weissman, "IoT Security Using Deception – Measuring Improved Risk Posture," 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2020, pp. 1-2.
30. W. Park and G. Ahn, "A Study on the Next Generation Security Control Model for Cyber Threat Detection in the Internet of Things (IoT) Environment," 2021 21st ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter), Ho Chi Minh City, Vietnam, 2021, pp. 213-217.
31. B. B. Sundaram, A. Pandey, V. Janga, D. A. Wako, A. S. Genale and P. Karthika, "IoT Enhancement with Automated Device Identification for Network Security," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 531-535.
32. R. Johari, I. Kaur, R. Tripathi and K. Gupta, "Penetration Testing in IoT Network," 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 2020, pp. 1-7.

33. A. Muhaimeen, K. Aadithiyaprasana, A. Ranjith, S. P. Sasirekha, R. Reshma and N. Mekala, "Enhancing IoT Security with Federated Deep Learning Techniques," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 1081-1087.
34. X. Zuo, X. Pang, P. Zhang, J. Zhang, T. Dong and P. Zhang, "A Security-aware Software-defined IoT Network Architecture," 2020 IEEE Computing, Communications and IoT Applications (ComComAp), Beijing, China, 2020, pp. 1-5.
35. O. Almazrouei, P. Magalingam, M. Kamrul Hasan, M. Almehrzi and A. Alshamsi, "Penetration Testing for IoT Security: The Case Study of a Wireless IP Security CAM," 2023 IEEE 2nd International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, 2023, pp. 1-5.
36. S. Bansal and V. K. Tomar, "Challenges & Security Threats in IoT with Solution Architectures," 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 2022, pp. 1-5.
37. W. Najib, S. Sulistyo and Widyawan, "Trust Based Security Model in IoT Ecosystem," 2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, 2022, pp. 195-199.
38. B. Kim, S. Yoon and Y. Kang, "Reinforcement of IoT Open Platform Security using PUF -based Device Authentication," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 1969-1971.
39. K. P. Singh, V. Rishiwal and P. Kumar, "Classification of Data to Enhance Data Security in Cloud Computing," 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-5.

40. K.Y. Lam, S. Mitra, F. Gondesen and X. Yi, "ANT-Centric IoT Security Reference Architecture—Security-by-Design for Satellite-Enabled Smart Cities," in IEEE Internet of Things Journal, vol. 9, no. 8, pp. 5895-5908, 15 April15, 2022.
41. M. Chu and Y. Song, "Analysis of network security and privacy security based on AI in IOT environment," 2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 2021, pp. 390-393.
42. J. Fan, Y. Xu and J. Ma, "Research on the Design of Network Security Architecture of Power Grid Enterprises," 2022 5th International Conference on Power and Energy Applications (ICPEA), Guangzhou, China, 2022, pp. 284-289.
43. N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," in IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2702-2733.
44. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин.
45. ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку».
46. Державні санітарні норми та правила «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу».
47. «ДСН 3.3.6.042-99 Санітарних норми мікроклімату виробничих приміщень».
48. ДБН 13.2.5-28-2006 «Природне і штучне освітлення».
49. ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку».
50. ДСТУ 12.1.005-88 «ССБП. Загальні санітарно-гігієнічні вимоги до повітря робочої зони».
51. ДБН В.2.5-28-2006 «Інженерне обладнання будинків і споруд. Природне і штучне освітлення».

52. ДСТУ Б В.2.5-82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом».
53. ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги».
54. НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою».
55. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин.
56. Environmental risk forecasting using hierarchy analysis and fuzzy set theory: міжнародна науково-практична конференція «І-й всеукраїнський з'їзд екологів»: Тези доповідей. Україна, м. Вінниця, 4-7 жовтня 2016. – 2016. – С.25.
57. Антипов В.В., Давидов Б.І., Тихончук В.С. Біологічна дія, нормування та захист від електромагнітних випромінювань. К.: Енергоатоміздат, 2002. - 177 с.
58. Філіппов Є.С. Вплив електромагнітних полів на біологічні об'єкти / Є.С. Філіппов, Є.Л. Ткачук // Львів. - 2018. -№1 - Том: 24. - С. 15-19.
59. Екологія та охорона навколишнього природного середовища: навч. посібник для вузів / В. С. Джигирей. - 6-те вид., випр. і доп. - К. : Знання, 2017. - 422 с.
60. Боротьба з шумом на виробництві: Довідник / Під ред. О. Я. Юдіна. – М: Машинобудування, 2015. – 297 с.

ДОДАТОК А

XML SCHEMA-ФАЙЛ ДЛЯ XML-ФАЙЛУ СКЛАДУ СИСТЕМИ

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2023/XMLSchema">
  <xs:element name="System" type="systemcomponents"/>
  <xs:complexType name="systemcomponents">
    <xs:sequence>
      <xs:element name="Subsystem" type="subsystemcomponents" minOccurs="1"
maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="subsystemcomponents">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Component" type="componentcomponents" minOccurs="1"
maxOccurs="1000"/>
    </xs:sequence>
    <xs:attribute name="ID" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:positiveInteger">
          <xs:minInclusive value="1"/>
          <xs:maxInclusive value="100"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="componentcomponents">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Object" type="objectcomponents" minOccurs="1" maxOccurs="1000"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
  <xs:complexType name="objectcomponents">
    <xs:choice minOccurs="2" maxOccurs="unbounded">
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Actuator" type="actuatorcomponents" minOccurs="0" maxOccurs="10"/>
      <xs:element name="Sensor" type="sensorcomponents" minOccurs="0" maxOccurs="10"/>
    </xs:choice>
    <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
  <xs:complexType name="sensorcomponents">
    <xs:choice minOccurs="1" maxOccurs="4">
      <xs:element name="Type">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Analog"/>
            <xs:enumeration value="Digital"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Min" type="xs:integer"/>
      <xs:element name="Max" type="xs:integer"/>
      <xs:element name="Average">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:integer">
              <xs:attribute name="Same" use="required">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
```

```
                    <xs:enumeration value="Yes"/>
                    <xs:enumeration value="No"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:choice>
    <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
  <xs:complexType name="actuatorcomponents">
    <xs:choice minOccurs="1" maxOccurs="2">
      <xs:element name="Type">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Switch"/>
            <xs:enumeration value="Action"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Average">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="Same" use="required">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="Yes"/>
                    <xs:enumeration value="No"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:choice>
    <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
        </xs:schema>
```

ДОДАТОК Б

XML SCHEMA-ФАЙЛ ДЛЯ XML-ФАЙЛУ "ІДЕАЛЬНОГО" СТАНУ СИСТЕМИ

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2023/XMLSchema">
  <xs:element name="System" type="systemcomponents"/>
  <xs:complexType name="systemcomponents">
    <xs:sequence>
      <xs:element name="Subsystem" type="subsystemcomponents" minOccurs="1"
maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" use="required"/>
    <xs:attribute name="DateFrom" type="xs:string" use="required"/>
    <xs:attribute name="DateTo" type="xs:string" use="required"/>
    <xs:attribute name="EverySec" type="xs:positiveInteger" use="required"/>
    <xs:attribute name="Round" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
  <xs:complexType name="subsystemcomponents">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Component" type="componentcomponents" minOccurs="1"
maxOccurs="1000"/>
    </xs:sequence>
    <xs:attribute name="ID" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:positiveInteger">
          <xs:minInclusive value="1"/>
          <xs:maxInclusive value="100"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="componentcomponents">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Object" type="objectcomponents" minOccurs="1" maxOccurs="1000"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
  <xs:complexType name="objectcomponents">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Sensor" type="sensorcomponents" minOccurs="0" maxOccurs="10"/>
      <xs:element name="Actuator" type="actuatorcomponents" minOccurs="0" maxOccurs="10"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
  <xs:complexType name="sensorcomponents">
    <xs:sequence>
      <xs:element name="Type">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Analog"/>
            <xs:enumeration value="Digital"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Values" type="sensorvaluescomponents"/>
      <xs:element name="Min" type="xs:positiveInteger"/>
      <xs:element name="Max" type="xs:positiveInteger"/>
      <xs:element name="Average">
        <xs:complexType>
```

```
        <xs:simpleContent>
          <xs:extension base="xs:positiveInteger">
            <xs:attribute name="Same" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="Yes"/>
                  <xs:enumeration value="No"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
</xs:complexType>
<xs:complexType name="actuatorcomponents">
  <xs:sequence>
    <xs:element name="Type">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Switch"/>
          <xs:enumeration value="Action"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Values" type="actuatorvaluescomponents"/>
    <xs:element name="Average">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="Same" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="Yes"/>
                  <xs:enumeration value="No"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
</xs:complexType>
<xs:complexType name="actuatorvaluescomponents">
  <xs:sequence>
    <xs:element name="Value">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="actuatorvaluetype">
            <xs:attribute name="DateTime" type="xs:dateTime" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="actuatorvaluetype">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ON"/>
    <xs:enumeration value="OFF"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="sensorvaluetype">
  <xs:restriction base="xs:integer"/>
</xs:simpleType>
<xs:complexType name="sensorvaluescomponents">
  <xs:sequence>
    <xs:element name="Value">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="sensorvaluetype">
            <xs:attribute name="DateTime" type="xs:dateTime" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
        </xs:schema>
```

ДОДАТОК В

XML SCHEMA-ФАЙЛ ДЛЯ XML-ФАЙЛУ ЗАГРОЗ "РОЗУМНОГО БЗНИНКУ"

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2023/XMLSchema">
  <xs:element name="Threats">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Object"  minOccurs="1" maxOccurs="10000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Sensor"   maxOccurs="1000">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Condition" minOccurs="1" maxOccurs="1000">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Threat" type="threatcomponents"  minOccurs="1"
maxOccurs="1000"/>
                        </xs:sequence>
                        <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
                        <xs:attribute name="Type" type="xs:positiveInteger" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Actuator"  maxOccurs="1000">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Condition" minOccurs="1" maxOccurs="1000">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Threat" type="threatcomponents"  minOccurs="1"
maxOccurs="1000"/>
                        </xs:sequence>
                        <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
                        <xs:attribute name="Type" type="xs:positiveInteger" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="Name" type="xs:string" use="required"/>
            <xs:attribute name="SubsystemID" type="xs:positiveInteger" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="threatcomponents">
    <xs:sequence>
      <xs:element name="Consequences" type="consequencescomponents"/>
      <xs:element name="Sources" type="sourcescomponents"/>
      <xs:element name="Causes" type="causescomponents"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>
    <xs:attribute name="Name" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="consequencescomponents">
    <xs:sequence>
```

```
    <xs:element name="Consequence" minOccurs="0" maxOccurs="100">
      <xs:complexType mixed="true">
        <xs:attribute name="ID" type="xs:positiveInteger"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="sourcescomponents">
  <xs:sequence>
    <xs:element name="Source" minOccurs="0" maxOccurs="100">
      <xs:complexType mixed="true">
        <xs:attribute name="ID" type="xs:positiveInteger"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="causescomponents">
  <xs:sequence>
    <xs:element name="Cause" minOccurs="0" maxOccurs="100">
      <xs:complexType mixed="true">
        <xs:attribute name="ID" type="xs:positiveInteger"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
        </xs:schema>
```

ДОДАТОК Г

ЛІСТИНГ Г.1 - МЕТОД ОТРИМАННЯ "ІДЕАЛЬНОГО" СТАНУ

```
private void Get_source_usual(bool now, string datefrom, string dateto)
        {
            DateTime d1 = DateTime.Now;//дата та час при натисканні кнопки для отримання
ідеального стану
             DateTime current = d1;
             DateTime next;
             DateTime d2 = d1;
             int tm = 0;
            string dates = "";//логи для методу
             string doc_path = "..\\..\\source_usual.xml";
             try
             {
                string system_name = textbox.Text;//дані з форми
                 string everysec = textbox1.Text;//
                 string round = textbox2.Text;//
                string datefrom_ = datefrom;//дата початку у форматі xml
                 datefrom_ = datefrom_.Replace(".", ":");
                 datefrom_ = datefrom_.Replace(" ", "T");
                string dateto_ = dateto;//дата закінчення у форматі xml
                 dateto_ = dateto_.Replace(".", ":");
                 dateto_ = dateto_.Replace(" ", "T");
                int current_it = 0;//номер ітерації

                int count = GetCount(datefrom, dateto, everysec);//розраховуємо кількість
необхідних ітерацій
                 dates += "count=" + count.ToString() + "\r\n";

                StreamWriter sw = File.CreateText(doc_path);//створюємо файл із кореневим тегом
і атрибутами
                 {
                     sw.WriteLine("<?xml version=\"1.0\" encoding=\"utf-8\"
standalone=\"no\"?>");
                     sw.WriteLine("<System Name=\"" + system_name + "\" DateFrom=\"" +
datefrom_ + "\" DateTo=\"" + dateto_ + "\" EverySec=\"" + everysec + "\" Round=\"" + round +
"\" xmlns:xsi=\"http://www.w3.org/2023/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\"schema source usual.xsd\">");
                     sw.WriteLine("</System>");
                 }
                 sw.Close();

                 string xpath;
                 XmlDocument doc = new XmlDocument();
                 String condition = null;
                 String date = null;
                 String time = null;
                 String subsystem_id = null;
                 String component_id = null;
                 String object_id = null;
                 String elem_tagname = null;
                 String elem_id = null;
                 String type = null;
                 String value = null;
                 String input_path = "";

                 if (!now)                              //Чекаємо початку збору даних
                 {
                     TimeSpan ts = Convert.ToDateTime(datefrom) - d1;
                     tm = Convert.ToInt32(Math.Ceiling(ts.TotalSeconds));
                     tm = tm * 1000;
```

```
                    dates = "wait=" + tm.ToString() + "msec" + "\r\n";
                    System.Threading.Thread.Sleep(tm);
                    dates += "after sleep time is " + DateTime.Now.ToString() + "\r\n";
                }

            while (current_it <= count)//поки поточна ітерація менша за кількість
запланованих ітерацій
                {
                   current = DateTime.Now;//поточний час дата
                   next = current.AddSeconds(Convert.ToDouble(everysec));//час дата для
наступної ітерації
                    dates += "current_it=" + current_it + "   current=" + current.ToString()
+ ",  next=" + next.ToString() + "\r\n";

                   if (current_it != 0)//якщо це не перша ітерація, то чекаємо початку
часу цієї ітерації, якщо було запущено відкладене отримання ідеального стану
                    {
                        TimeSpan ts = next - current;
                        tm = (Convert.ToInt32(Math.Floor(ts.TotalSeconds)) - 2) * 1000;
                        System.Threading.Thread.Sleep(tm);
                        dates += "after sleep " + tm.ToString() + "msec   in  current_it=" +
current_it + ",   time is " + DateTime.Now.ToString() + "\r\n";
                    }

                    current_it++;
                   input_path = GenerateFile();//генеруємо файл

                    foreach (string line in File.ReadLines(input_path,
Encoding.UTF8))//разбор файла
                    {
                        if (line[0] != '<')
                        {
                            XmlNode root_for_element = null;
                            XmlNode root_for_object = null;
                            XmlNode root_for_component = null;
                            XmlNode root = null;
                            XmlElement element_for_value = null;
                            doc.Load(doc_path);
                            XDocument xdoc = XDocument.Load(doc_path);
                            int s_id;
                            int c_id;
                            int o_id;
                            int ind;
                            string tmp = "";

                            ind = line.IndexOf("/");
                            date = line.Substring(0, ind);
                            tmp = line.Remove(0, ind + 1);

                            ind = tmp.IndexOf("/");
                            time = tmp.Substring(0, ind);
                            tmp = tmp.Remove(0, ind + 1);

                            ind = tmp.IndexOf("/");
                            subsystem_id = tmp.Substring(0, ind);
                            tmp = tmp.Remove(0, ind + 1);

                            ind = tmp.IndexOf("/");
                            component_id = tmp.Substring(0, ind);
                            tmp = tmp.Remove(0, ind + 1);

                            ind = tmp.IndexOf("/");
                            object_id = tmp.Substring(0, ind);
```

```
                            tmp = tmp.Remove(0, ind + 1);

                            ind = tmp.IndexOf("/");
                            elem_tagname = tmp.Substring(0, ind);
                            if (elem_tagname == "A")
                            { elem_tagname = "Actuator"; }
                            else if (elem_tagname == "S")
                            { elem_tagname = "Sensor"; }
                            tmp = tmp.Remove(0, ind + 1);

                            ind = tmp.IndexOf("/");
                            elem_id = tmp.Substring(0, ind);
                            tmp = tmp.Remove(0, ind + 1);

                            ind = tmp.IndexOf("/");
                            type = tmp.Substring(0, ind);
                            tmp = tmp.Remove(0, ind + 1);

                            value = tmp;

                            s_id = xdoc.Descendants("Subsystem").Where(x =>
x.Attribute("ID").Value == subsystem_id).Count();
                            c_id = xdoc.Descendants("Component").Where(x =>
x.Attribute("ID").Value == component_id).Count();
                            o_id = xdoc.Descendants("Object").Where(x =>
x.Attribute("ID").Value == object_id).Count();


                           if (s_id != 0 && c_id != 0 && o_id != 0)//створення елемента в
існуючій підсистемі, об'єкті та компоненті
                            {
                                condition = "all_old";
                            }
                           else if (s_id == 0)//усе нове, такої підсистеми ще немає у файлі
                            {
                                condition = "all_new";
                            }
                           else if (s_id != 0 && c_id == 0)//створення нового компонента і
об'єкта, в існуючій підсистемі
                            {
                                condition = "component_new";
                            }
                           else if (s_id != 0 && c_id != 0 && o_id == 0)//створення нового
об'єкта, в існуючому компоненті та підсистемі
                            {
                                condition = "object_new";
                            }

                            switch (condition)
                            {
                                case "1"://////actuator or sensor
                                    {
                                       //додавання нового елемента з одним значенням
                                        if (xdoc.Descendants(elem_tagname).Where(x =>
x.Attribute("ID").Value == elem_id).Count() == 0)
                                        {
                                            XmlElement elem =
doc.CreateElement(elem_tagname);

                                            elem.SetAttribute("ID", elem_id);
                                            root_for_element.AppendChild(elem);

                                            XmlElement elem1 = doc.CreateElement("Type");
                                            string type_ = "";
```

```
                    if (elem_tagname == "Actuator")//actuator
                    {
                        type_ = Actuator_types[type];
                    }
                    else if (elem_tagname == "Sensor")//sensor
                    {
                        type_ = Sensor_types[type];
                    }
                    elem1.InnerText = type_;
                    elem.AppendChild(elem1);

                    XmlElement elem2 = doc.CreateElement("Values");
                    elem.AppendChild(elem2);
                    element_for_value = elem2;

                    XmlElement elem3 = doc.CreateElement("Value");
                    elem3.SetAttribute("DateTime", date + "T" +
time);

                    elem3.InnerText = value;
                    elem2.AppendChild(elem3);
                }
                else //Додавання наступних отриманих значень
елемента
                {
                    xpath = "System/Subsystem[@ID='" + subsystem_id
+ "']/Component[@ID='" + component_id + "']/Object[@ID='" + object_id + "']/" + elem_tagname
+ "[@ID='" + elem_id + "']/Values";
                    XmlElement elem4 = doc.CreateElement("Value");
                    elem4.SetAttribute("DateTime", date + "T" +
time);
                    elem4.InnerText = value;
                    doc.SelectSingleNode(xpath).AppendChild(elem4);
                }
                doc.Save(doc_path);
                break;
            }
            case "all_old":///підсистема, компонент і об'єкт існують
            {
                xpath = "System/Subsystem[@ID='" + subsystem_id +
"']/Component[@ID='" + component_id + "']/Object[@ID='" + object_id + "']";
                root_for_element = doc.SelectSingleNode(xpath);
                goto case "1";
            }
            case "object_new":
            {
                if (root_for_object == null)//якщо сюди прийшли не з
"component_new", то отримуємо компонент, у якому будемо створювати об'єкт
                {
                    xpath = "System/Subsystem[@ID='" + subsystem_id
+ "']/Component[@ID='" + component_id + "']";
                    root_for_object = doc.SelectSingleNode(xpath);
                }
                XmlElement elem0 = doc.CreateElement("Object");
                elem0.SetAttribute("ID", object_id);
                root_for_object.AppendChild(elem0);

                XmlElement elem01 = doc.CreateElement("Name");
                elem0.AppendChild(elem01);

                root_for_element = elem0;
                goto case "1";

            }
```

```
                                case "component_new":
                                    {
                                       if (root_for_component == null)//якщо сюди прийшли не
із "all_new", то отримуємо підсистему, в якій будемо створювати компонент
                                        {
                                            xpath = "System/Subsystem[@ID='" + subsystem_id
+ "']";
                                            root_for_component =
doc.SelectSingleNode(xpath);
                                        }

                                        XmlElement elem1 = doc.CreateElement("Component");
                                        elem1.SetAttribute("ID", component_id);
                                        root_for_component.AppendChild(elem1);
                                        XmlElement elem11 = doc.CreateElement("Name");
                                        elem1.AppendChild(elem11);
                                        root_for_object = elem1;
                                        goto case "object_new";

                                    }
                                case "all_new":
                                    {
                                        xpath = "System";
                                        root = doc.SelectSingleNode(xpath);
                                        XmlElement elem00 = doc.CreateElement("Subsystem");
                                        elem00.SetAttribute("ID", subsystem_id);
                                        root.AppendChild(elem00);
                                        XmlElement elem001 = doc.CreateElement("Name");
                                        elem00.AppendChild(elem001);
                                        root_for_component = elem00;
                                        goto case "component_new";
                                    }
                            }
                        }

                    }
                }

                d2 = DateTime.Now;
                dates += "after get data time is " + d2.ToString() + "\r\n";
               DeleteGeneratedFiles();//видаляємо згенеровані файли

                ////////////////////////////////////////
                ////////////////////////////////average

                String average;
                doc.Load(doc_path);
                XDocument xdoc2 = XDocument.Load(doc_path);
                var all_sen = xdoc2.Descendants("Sensor");
                var all_act = xdoc2.Descendants("Actuator");

                foreach (XElement sen in all_sen)
                {
                    var obj = sen.Descendants("Value")
                              .Select(y => Convert.ToInt32(y.Value));//список усіх
значень сенсора

                    average = Math.Round(obj.Average(), Convert.ToInt32(round)).ToString();
                    int min = obj.Min();
                    int max = obj.Max();
                    sen.Add(new XElement("Min", min.ToString()));
                    sen.Add(new XElement("Max", max.ToString()));
```

```
                    XElement aver = new XElement("Average", average.ToString());
                    sen.Add(aver);

                    if (min == max)
                    {
                        aver.SetAttributeValue("Same", "Yes");
                    }
                    else
                    {
                        aver.SetAttributeValue("Same", "No");
                    }
                }

                foreach (XElement act in all_act)
                {
                    var obj = act.Descendants("Value")
                                .Select(y => y.Value);//список усіх значень активатора

                    int on = 0;
                    bool same = false;
                    string tmp = "";
                    int obj_count = obj.Count();

                    foreach (string val in obj)
                    {
                        if (val == "1")
                        {
                            on++;
                        }
                        tmp = val;
                    }

                    if (on == obj_count || on == 0)
                    {
                        same = true;
                    }


                    XElement aver = new XElement("Average");
                    act.Add(aver);

                    if (same)
                    {
                        aver.SetAttributeValue("Same", "Yes");
                        aver.Value = (tmp == "1") ? "100" : "0";

                    }
                    else
                    {
                        aver.SetAttributeValue("Same", "No");
                        aver.Value = (Math.Round(Convert.ToDecimal(on * 100 /
obj_count))).ToString();
                        //MessageBox.Show( "obj.count=" + obj_count.ToString() + ", on=" +
on + ", aver value="+aver.Value.ToString());
                    }
                }
                xdoc2.Save(doc_path, SaveOptions.None);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Exception in get source_usual.xml=" + ex.Message + "\r\n" +
"Source=" + ex.Source + "\r\n" + "StackTrace=" + ex.StackTrace);
            }


            finally
            {
                MessageBox.Show("done everything");
                DateTime d3 = DateTime.Now;
                dates += "after all time is " + d3.ToString();
                //MessageBox.Show(dates);
                //MessageBox.Show("d1=" + d1.ToString() + "\r\n" + "current=" +
current.ToString() + "\r\n" + "d2=" + d2.ToString() + "\r\n" + "d3=" + d3.ToString());
            }

                }
```

ДОДАТОК Д

ЛІСТИНГ Д.1 - МЕТОД ДОДАВАННЯ ЕЛЕМЕНТА ЛІСТИНГ Д.2 - МЕТОД ДОДАВАННЯ ЕЛЕМЕНТА

```
private void Add()////додавання нових елементів
        {
        try
            {
                bool res_getid = getID();//false - помилка під час отримання ID для нових
елементів

                if (res_getid == false || (is_new_component == null || is_new_object == null
|| /*is_sensor*/element_tagname == null ||
                    (is_new_component == true && is_new_object == false) ||
                   subsystem_id == null || subsystem_id == "" ||
                    component_id == null || component_id == "" ||
                    object_id == null || object_id == ""  ||
                    element_id == null || element_id == "" || element_type == null ||
element_type == "" || element_average == null || element_average == ""))
                {
                   MessageBox.Show("Помилка під час додавання елемента:" + "\r\n" +
                   "тип елемента = " + element_type + "\r\n" +
                    "average = " + element_average);
                }


                else
                {
                   //вираз для кореневого елемента - нижній існуючий, у який
буде додаватися новий
                    string xpath;
                    string doc_path=file_path;
                    XmlDocument doc = new XmlDocument();

                    String condition=null;
                    XmlNode root_for_element=null;
                    XmlNode root_for_object=null;
                    XmlNode root_for_component = null;
                    XmlNode root=null;

                   if (is_new_component == false && is_new_object == false)//створення
елемента в існуючих об'єкті та компоненті
                    {
                        condition = "all_old";
                        doc.Load(doc_path);
                    }
                    else if (doc.SelectSingleNode("//Subsystem[@ID='" + subsystem_id + "']")
== null)//якщо немає такої підсистеми
                    {
                        condition = "all_new";
                       if (!File.Exists(file_path))//створення всього, такого файлу ще немає
                        {
                            StreamWriter sw = File.CreateText(doc_path);
                            {
                                sw.WriteLine("<?xml version=\"1.0\" encoding=\"utf-8\"
standalone=\"no\"?>");
                                sw.WriteLine("<System Name=\"system_name\"
xmlns:xsi=\"http://www.w3.org/2023/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\"schema source.xsd\">");
                                sw.WriteLine("</System>");
                            }
                            sw.Close();
```

```
                    } doc.Load(doc_path);

                }
                else if (/*is_new_subsystem == false && */is_new_component ==
true)//створення нового компонента, тобто і нового об'єкта теж
                {
                    condition = "component_new";
                    doc.Load(doc_path);
                }
                else if (/*is_new_component == false &&*/ is_new_object ==
true)//створення нового об'єкта, але в існуючому компоненті
                {
                    condition = "object_new";
                    doc.Load(doc_path);
                }



                    switch (condition)
                    {

                        case "1":
                            {
                                //Create a new node - actuator or sensor
                                XmlElement elem = doc.CreateElement(element_tagname);
                                elem.SetAttribute("ID", element_id);
                                //Add the node to the document.
                                root_for_element.AppendChild(elem);

                                XmlElement elem1 = doc.CreateElement("Type");
                                elem1.InnerText = element_type;
                                elem.AppendChild(elem1);

                                XmlElement elem2 = doc.CreateElement("Average");
                                elem2.SetAttribute("Same", "No");
                                elem2.InnerText = element_average;
                                elem.AppendChild(elem2);
                                if (element_tagname == "Sensor")
                                {
                                    XmlElement elem3 = doc.CreateElement("Max");
                                    elem3.InnerText = element_max;
                                    elem.AppendChild(elem3);
                                    XmlElement elem4 = doc.CreateElement("Min");
                                    elem4.InnerText = element_min;
                                    elem.AppendChild(elem4);
                                }
                                doc.Save(doc_path);
                                MessageBox.Show("Готово");
                                break;
                            }
                        case "all_old":
                            {
                                xpath = "//Subsystem[@ID='" + subsystem_id +
"']/Component[@ID='" + component_id + "']/Object[@ID='" + object_id + "']";
                                root_for_element = doc.SelectSingleNode(xpath);
                                goto case "1";
                            }
                        case "object_new":
                            {
                               if (root_for_object == null)//якщо сюди прийшли не з
"component_new", то отримуємо компонент, у якому будемо створювати об'єкт
                                {
```

```
                                    xpath = "//Subsystem[@ID='" + subsystem_id +
"']/Component[@ID='" + component_id + "']";
                                    root_for_object = doc.SelectSingleNode(xpath);
                                }
                                XmlElement elem0 = doc.CreateElement("Object");
                                elem0.SetAttribute("ID", object_id);
                                root_for_object.AppendChild(elem0);

                                XmlElement elem01 = doc.CreateElement("Name");
                                elem01.InnerText = object_name;
                                elem0.AppendChild(elem01);

                                root_for_element = elem0;
                                goto case "1";

                            }
                        case "component_new":
                            {
                               if (root_for_component == null)//якщо сюди прийшли не з
"all_new", то отримуємо підсистему, в якій будемо створювати компонент
                                {
                                    xpath = "//Subsystem[@ID='" + subsystem_id + "']";
                                    root_for_component = doc.SelectSingleNode(xpath);
                                }
                                XmlElement elem000 = doc.CreateElement("Component");
                                elem000.SetAttribute("ID", component_id);
                                root_for_component.AppendChild(elem000);
                                XmlElement elem0001 = doc.CreateElement("Name");
                                elem0001.InnerText = component_name;
                                elem000.AppendChild(elem0001);

                                root_for_object = elem000;
                                goto case "object_new";

                            }
                        case "all_new":
                            {
                                xpath = "System";
                                root = doc.SelectSingleNode(xpath);

                                XmlElement elem00 = doc.CreateElement("Subsystem");
                                elem00.SetAttribute("ID", subsystem_id);
                                root.AppendChild(elem00);
                                XmlElement elem001 = doc.CreateElement("Name");
                                elem001.InnerText = subsystem_name;
                                elem00.AppendChild(elem001);

                                root_for_component = elem00;
                                goto case "component_new";
                            }
                    }

                }
            this.Close();

        }
         catch (Exception ee)
         {
            MessageBox.Show("Помилка додавання: "+"\r\n" +
ee.Message+"\r\n"+ee.StackTrace+"\r\n"+ee.Source);
         }
               }
```

ДОДАТОК Е

ЛІСТИНГ Е.1 - МЕТОД МОНІТОРИНГУ ДАНИХ

```
private static void Monitoring()//потік для моніторингу даних у data.txt
        {
                 string data = generated_files[0];
                int error = 0;//код помилки = номеру даних або =100, якщо не збігається
кількість елементів у файлі
                string tmp = "";//тимчасова змінна для зберігання окремих даних з
рядка
                int tmp_num = 1;//номер даних: 1-date, 2-time, 3-id subsystem, 4-id
component, 5-id object, 6-element(S or A), 7-id element, 8-element type, 9-element value
                int row = -1;//індекс рядка
                int c_ind = -1;//індекс символу в рядку
                string elemtagname="";//A-активатор або S-сенсор
                string elemid = "";//id елемента для перевірки типу елемента
                string xpath = "";//шлях для перевірки вкладеності
                 var file = File.ReadAllLines(data);
                 List<string> list1 = new List<string>(file);

                if (list1.Count != elements_count)//якщо не збігається кількість елементів
                 {
                     error = 100;
                     WriteLog(error, -1, data);
                     int ind = 0;
                     string subsystem_id;
                     string component_id;
                     string object_id;
                     string s_id;
                     string a_id;
                     string tag;
                    //списки з елементами системи, визначеними у складі системи
                     List<string> act_source = Actuators_in_source;
                     List<string> sen_source = Sensors_in_source;
                     List<string> sub_source = Subsystems_in_source;
                     List<string> com_source = Components_in_source;
                     List<string> obj_source = Objects_in_source;
                     /////
                    //перевіряємо кількість підсистем (кількісний аналіз даних)
                     foreach (string str in list1)
                     {
                         tmp = "";
                         ind = str.IndexOf("/");
                         tmp = str.Remove(0, ind + 1);//delete date
                         ind = tmp.IndexOf("/");
                         tmp = tmp.Remove(0, ind + 1);//delete time
                         ind = tmp.IndexOf("/");
                         subsystem_id = tmp.Substring(0, ind);//subsystem id
                         tmp = tmp.Remove(0, ind + 1);//delete subsystem id
                         ind = tmp.IndexOf("/");
                         component_id = tmp.Substring(0, ind);//component_id
                         tmp = tmp.Remove(0, ind + 1);
                         ind = tmp.IndexOf("/");
                         object_id = tmp.Substring(0, ind);//object_id
                         tmp = tmp.Remove(0, ind + 1);
                         ind = tmp.IndexOf("/");
                         tag = tmp.Substring(0, ind);
                         if (tag == "A")
                         {
                             tmp = tmp.Remove(0, ind + 1);
                             ind = tmp.IndexOf("/");
                             a_id = tmp.Substring(0, ind);
```

```
                    tmp = tmp.Remove(0, ind + 1);
                    act_source.Remove(a_id);
                }
                else
                {
                    tmp = tmp.Remove(0, ind + 1);
                    ind = tmp.IndexOf("/");
                    s_id = tmp.Substring(0, ind);
                    tmp = tmp.Remove(0, ind + 1);
                    sen_source.Remove(s_id);
                }
                sub_source.Remove(subsystem_id);
                sub_source.Remove(component_id);
                obj_source.Remove(object_id);
            }
            if (sub_source.Count > 0)
            {
                foreach (string s in sub_source)
                {
                   WriteLog(103, -1, data, Convert.ToInt32(s));//запис помилки в
лог
                }
            }
           ///////Перевіряємо кількість компонентів//
            else
            {
                if (com_source.Count > 0)
                {
                    foreach (string s in com_source)
                    {
                       WriteLog(104, -1, data, Convert.ToInt32(s));//запис помилки
у лог
                    }
                }
               else///////Перевіряємо кількість objects//
                {
                    if (obj_source.Count > 0)
                    {
                        foreach (string s in obj_source)
                        {
                            WriteLog(105, -1, data, Convert.ToInt32(s));
                        }
                    }
                    else//////Actuators and Sensors
                    {
                        if (act_source.Count > 0)
                        {
                            foreach (string s in act_source)
                            {
                                WriteLog(1072, -1, data, Convert.ToInt32(s));
                                int th_id = GetThreatID("Actuator",s,100);
                                Bad_actuators.Add(Convert.ToInt32(s),
Get_threat_info(th_id, 1072, data.Substring(10, 15)));//додаємо до списку активаторів із
загрозами
                            }
                        }
                        if (sen_source.Count > 0)
                        {
                            foreach (string s in sen_source)
                            {
                                WriteLog(1071, -1, data, Convert.ToInt32(s));
                                int th_id = GetThreatID("Sensor",s,100);
```

```
                                    Bad_sensors.Add(Convert.ToInt32(s),
Get_threat_info(th_id, 1071, data.Substring(10,15)));//додаємо до списку сенсорів із
загрозами
                                }
                            }
                        }
                    }
                }

                else
                {
                    foreach (string str in list1)
                    {
                        error = 0;
                        tmp = "";
                        tmp_num = 1;
                        c_ind = -1;
                        elemtagname = "";
                        elemid = "";
                        row++;
                        xpath = "";

                        foreach (char c in str)
                        {
                            c_ind++;

                            if (!c.Equals('/'))
                            {
                                tmp += c;
                            }
                            if (c.Equals('/') || c_ind == str.Length - 1)
                            {
                                switch (tmp_num)
                                {
                                    case 6:
                                        elemtagname = tmp;
                                        break;
                                    case 3:
                                        xpath += "//Subsystem[@ID='" + tmp + "']";
                                        break;
                                    case 4:
                                        xpath += "//Component[@ID='" + tmp + "']";
                                        break;
                                    case 5:
                                        xpath += "//Object[@ID='" + tmp + "']";
                                        break;
                                    case 7:
                                        elemid = tmp;
                                        break;
                                }
                                error = CheckDataStructure(tmp_num, tmp);//перевіряємо
правильність структури кожного рядка в отриманих даних
                                if (error != 0)//помилка в структурі отриманих даних
                                {
                                    WriteLog(error, row, data);
                                }
                                else if (tmp_num == 3 || tmp_num == 4 || tmp_num == 5 ||
tmp_num == 7 || tmp_num == 8 || tmp_num == 9)//немає помилки в структурі, перевіряємо склад
                                {
                                   if (tmp_num != 9)//якщо це не дані з елемента, а id

                                    {
```

```
                                    error = CheckSystemStructure(tmp_num, tmp,
elemtagname, elemid, xpath);//перевіряємо чи збігається склад системи
                                }
                                if (error != 0)//помилка у складі
                                {
                                    WriteLog(error, row, data);
                                }
                                else if (error == 0 && tmp_num == 9)//якщо це дані з
елемента
                                {
                                    error = CheckValue(tmp, elemtagname,
elemid);//перевіряємо дані елемента на відповідність обмеженням//метод повертає
%відхилення або 0
                                   if (error != -1)//помилка в даних елемента
                                    {
                                        if (elemtagname == "S")
                                        {
                                            WriteLog(99, row, data,
Convert.ToInt32(elemid));
                                            error = GetThreatID(elemtagname, elemid,
error);//threat id
                                            if
(Bad_sensors.ContainsKey(Convert.ToInt32(elemid)))
                                            {

Bad_sensors.Remove(Convert.ToInt32(elemid));
                                            }
                                            Bad_sensors.Add(Convert.ToInt32(elemid),
Get_threat_info(error, 99, data.Substring(10,15)));//додаємо до списку сенсорів із загрозами
                                        }
                                        if (elemtagname == "A")//actuator
                                        {
                                            error = GetThreatID(elemtagname, elemid,
error);//threat id
                                        }
                                       if (error != -1)//якщо виявлено загрозу
                                        {
                                            WriteLog(error, -1, "",
Convert.ToInt32(elemid));
                                        }
                                    }
                                   else//якщо немає помилок у даних
                                    {
                                       File.Delete(data);//видаляємо файл
                                    }
                                }
                            }
                            tmp = "";
                            tmp_num++;
                        }
                    }
                }
            }
            generated_files.RemoveAt(0);
            if (Bad_actuators.Count != 0 || Bad_sensors.Count != 0)
            {
                //updatetable();
                Write_dictionary_to_list();
                Bad_sensors.Clear();
                Bad_actuators.Clear();
            }

              }
```