НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 004.9:728-021.431

«ПОГОДЖЕНО»
Декан факультету
інформаційних технологій
Глазунова О.Г., д.п.н., професор_
_____
____ _____ 202_ р.

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»
Завідувач кафедри комп'ютерних наук
Голуб Б.Л., к.т.н., доцент_____
_____
____ _____ 202_ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему __Система моніторингу периметру безпеки розумного будинку з мобільного пристрою_____

Спеціальність _____121 - Інженерія програмного забезпечення_____

_____

(код і назва)

Освітня програма ____Програмне забезпечення інформаційних систем__

_____

(назва)

Орієнтація освітньої програми _____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми
_____к.т.н., доцент_____ _____ ___Голуб Б. Л._____
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи
_____ _____ _____Даков С. Ю.____
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ _____Прокіпчук О.В._____
(підпис) (ПІБ студента)

КИЇВ-2023

НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) _____інформаційних технологій_____

ЗАТВЕРДЖУЮ
Завідувач кафедри_____
____комп'ютерних наук_____
___к.т.н., доцент_____ _____ Голуб Б. Л._
(науковий ступінь, вчене звання) (підпис) (ПІБ)
"____" _____ ___20____року

З А В Д А Н Н Я
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

_____
(прізвище, ім'я, по батькові)
Спеціальність _____121 - Інженерія програмного забезпечення_____
(код і назва)
Освітня програма ____Програмне забезпечення інформаційних систем_____
(назва)
Орієнтація освітньої програми_____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)
Тема магістерської кваліфікаційної роботи_____Система моніторингу периметру безпеки розумного будинку з мобільного пристрою_____
_____
затверджена наказом ректора НУБіП України від "____"_____ 20___р. №_____
Термін подання завершеної роботи на кафедру ___
(рік, місяць, число)
Вихідні дані до магістерської кваліфікаційної роботи _____
_____
_____
_____
Перелік питань, що підлягають дослідженню:
1._____
2._____
3._____
Перелік графічного матеріалу (за потреби) _____
_____

Дата видачі завдання "____"_____20____ р.

Керівник магістерської кваліфікаційної роботи _____ __Даков_С.Ю._
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ _____Прокіпчук О.В._____
(підпис) (прізвище та ініціали студента)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CSS – Cascading Style Sheets

DM – Data Mining

ERD – Essence Relation Diagramming

ETL - Extract Transform Load

HTML – HyperText Markup Language

JS – JavaScript

OLAP – Online analytical processing

SADT – Structured Analysis and Design Technique

SASS – Syntactically Awesome Style Sheets

SSAS – SQL Server Analysis Services

SSIS – SQL Server Integration Services

UML – Unified Modeling Language

БД – база даних

ІТ – інформаційні технології

СУБД – система управління базою даних

ВСТУП

Актуальність

Наша сучасність повністю залежить від технологій, які зараз надзвичайно швидко розвиваються. Вони відіграють важливу роль майже в усіх сферах нашої діяльності та допомагають нам у повсякденному житті майже на кожному етапі. Майже кожна галузь зараз автоматизована, з використанням роботів, дронів, контролерів, датчиків, а також сучасного програмного забезпечення для більш комфортного, швидкого та якісного виробництва чи надання послуг.

Розумний будинок – одна з найактуальніших та найцікавіших інновацій у сучасній будівельній галузі. Ця концепція дозволяє будинкам бути по-справжньому "розумними", інтегруючи цілий ряд технологій і систем автоматизації, які підвищують комфорт, безпеку, ефективність і екологічність житла. Розумний дім став символом майбутнього, де технології допомагають людям у повсякденному житті.

Основна ідея розумного будинку полягає в інтеграції різних систем, таких як освітлення, опалення, кондиціонування, безпека та мережеві технології, в єдину централізовану систему, якою можна керувати зі смартфона, планшета або комп'ютера. Це дозволяє власникам контролювати свій будинок з будь-якої точки світу, віддалено налаштовувати параметри та отримувати інформацію в режимі реального часу.

Однією з головних переваг розумних будинків є економія енергії. Системи розумного будинку можуть автоматично регулювати споживання енергії відповідно до потреб домогосподарства та передбачуваного графіку. Наприклад, опалення та кондиціонер можна вимкнути, коли нікого немає вдома, або відрегулювати температуру відповідно до погоди.

Крім того, розумні будинки забезпечують вищий рівень безпеки. Вони обладнані системами відеоспостереження, відеодомофонами та сигналізацією, яка сповіщає власників про підозрілі події. Крім того, системи контролю доступу забезпечують безпечний вхід і вихід.

Розумні будинки також сприяють підвищенню зручності життя. Багато щоденних завдань, таких як приготування кави або регулювання освітлення, можуть бути автоматизовані. Крім того, інтеграція голосових помічників дозволяє власникам керувати будинком за допомогою голосових команд.

Об’єкт та предмет дослідження

Об’єктом дослідження є процеси дотримання безпеки на території розумного будинку.

Предметом дослідження виступає процес моніторингу, який допоможе надалі покращити безпекову ситуації на периметрі розумного будинку.

Мета дослідження

Метою дослідження є визначення корисності використання технологій OLAP і Data Mining для підвищення ефективності моніторингу безпеки розумного будинку.

Наукова новизна

1. Досліджено технології OLAP і Data Mining.
2. Розроблена архітектура системи
3. Для підсистеми аналізу було використано технології OLAP та Data Mining

Апробація результатів дослідження

Структура магістерської роботи

Магістерська кваліфікаційна робота складається з 4-х розділів.

Розділ 1. Проведений аналіз предметної області, формулювання технічного завдання, а також проаналізовано наявні рішення.

Розділ 2. Було проаналізовано систему та проведено моделювання системи використовуючи структурно-функціональне та об’єктно-орієнтоване моделювання.

Розділ 3. Продемонстровано розробку архітектури системи моніторингу, структура опертивної бази даних, сховища даних, розгортання багатовимірного

кубу на основі сховища даних, реалізацію алгоритмів для проведення аналізу даних.

Розділ 4. У даному розділі описано впровадження розроблюваної системи, продемонстровано результат роботи аналітичного модуля з використанням алгоритмів і сформовані висновки щодо їхніх результатів.

Матеріал магістерської кваліфікаційної роботи містить:

- кількість ілюстрацій – 31;
- кількість таблиць – 2;
- кількість використаних джерел – 30.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз процесу моніторингу безпеки розумного будинка

Розумний дім – це житло, в якому використовуються підключені до Інтернету пристрої для дистанційного моніторингу та управління приладами і системами, такими як освітлення та опалення.

Технологія розумного дому, яку також часто називають домашньою автоматизацією або домотикою (від латинського слова domus - дім), забезпечує власникам будинків безпеку, комфорт, зручність та енергоефективність, дозволяючи їм керувати розумними пристроями, часто за допомогою програми для розумного дому на смартфоні або іншому мережевому пристрої.

Будучи частиною інтернету речей (IoT), системи та пристрої розумного дому часто працюють разом, обмінюючись між собою даними про використання споживачами та автоматизуючи дії на основі вподобань власників будинків.

Всі пристрої такі як освітлення, термостати, системи безпеки та побутова техніка контролюються головним контролером домашньої автоматизації, який часто називають хабом розумного будинку. Хаб – це апаратний пристрій, який виступає в ролі центральної точки системи "розумного будинку" і може сприймати, обробляти дані та передавати їх бездротовим способом. Він об'єднує всі розрізнені додатки в єдину програму розумного будинку, якою власники будинків можуть керувати дистанційно. Прикладами хабів для розумного будинку є Amazon Echo, Google Home і Wink Hub. У той час як багато продуктів розумного будинку використовують Wi-Fi і Bluetooth для підключення до мережі розумного будинку, інші залежать від бездротових протоколів, таких як Zigbee або Z-Wave.

Нижче на рис. 1.1 наведено архітектуру системи розумного будинку, де ми можемо побачити, що у системі наявні декілька хабів, тобто центральних точок, які обробляють події, вимірювання та інше з різних пристроїв, датчиків. Для зручності та ефективності системи розумного будинку розділяють декілька

хабів, що виконують різні функції, як от до прикладу система освітлення, безпеки та моніторинг навколишнього середовища. Кожна з цих центральних точок оброблює та передає дані, які надалі можуть опрацьовуватись на сервері певним сервісом або набором сервісів. Усе це дозволяє підвищити ефективність роботи існуючої системи з метою можливого покращення комфорту та безпеки.

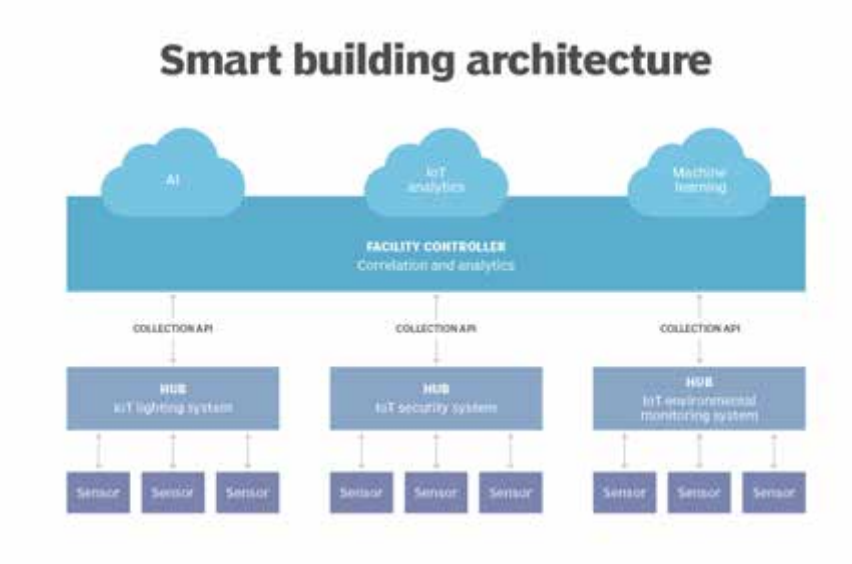

Рис. 1.1 Архітектура системи розумного будинку

Пристрої розумного будинку можна запрограмувати на виконання певних розкладів або команд, або ж налаштувати їх так, щоб вони реагували на голосові команди через домашніх помічників, таких як Amazon Alexa або Google Assistant. Наприклад, розумний термостат може вивчати звички власника будинку і автоматично регулювати температуру відповідно до його розкладу.

Хоча кожен розумний будинок – це розумна будівля, не кожна розумна будівля це розумний будинок. Корпоративні, комерційні, промислові та житлові будівлі всіх форм і розмірів в тому числі офіси, хмарочоси, багатоквартирні будинки і багатоквартирні офіси і житлові будинки впроваджують технології Інтернету речей для підвищення ефективності будівель, зниження витрат на енергію і шкоди навколишньому середовищу, забезпечення безпеки і підвищення рівня задоволеності мешканців.

Багато з тих самих інтелектуальних технологій, що використовуються в "розумному будинку", також застосовуються в технологіях "розумної будівлі", включаючи освітлення, енергоспоживання, опалення та кондиціонування, а також системи безпеки і доступу до будівлі.

Наприклад, "розумна" будівля може зменшити витрати на електроенергію за допомогою датчиків, які визначають кількість мешканців у приміщенні. Температура може автоматично регулюватися, вмикаючи прохолодне повітря, якщо датчики виявляють повну конференц-залу, або вимикаючи опалення, якщо всі в офісі пішли додому на цілий день.

Розумні будівлі також можуть підключатися до розумної мережі. У цьому випадку компоненти "розумної" будівлі та електромережа можуть розмовляти і слухати один одного. Ця технологія дозволяє більш ефективно управляти розподілом енергії, проводити технічне обслуговування і швидше реагувати на перебої в електропостачанні.

Окрім цих переваг, "розумні" будівлі можуть надавати власникам і менеджерам будівель переваги прогнозованого технічного обслуговування. Наприклад, прибиральники можуть поповнювати запаси в туалетах, коли датчики використання мила або паперових рушників показують, що вони закінчилися. Технічне обслуговування і збої в роботі холодильних систем, ліфтів і систем освітлення також можна передбачити заздалегідь.

Плюси і мінуси розумного будинку

Розумна технологія пропонує безліч переваг, починаючи від зручності використання побутових приладів, таких як пральна машина, під час роботи, і закінчуючи комфортом дистанційного регулювання термостата в холодний зимовий день.

До загальних переваг розумного будинку можна віднести наступні:

- Забезпечує впевненість. Домовласники можуть дистанційно контролювати своє житло, запобігаючи таким небезпекам, як забута кавоварка або незамкнені вхідні двері.

- Враховує вподобання користувачів для зручності. Наприклад,

користувачі можуть запрограмувати відчинення гаражних воріт, увімкнення світла, розпалювання каміна та відтворення улюбленої музики, коли вони повертаються додому.

- Підвищує ефективність. Замість того, щоб залишати кондиціонер увімкненим на весь день, система розумного дому може вивчати поведінку власників житла, щоб забезпечити охолодження будинку до їхнього повернення додому.

- Заощаджує ресурси та гроші. Завдяки розумній системі поливу газон поливається тільки тоді, коли це необхідно, і саме тією кількістю води, яка необхідна. Завдяки пристроям домашньої автоматизації та розумному налаштуванню системи енергія, вода та інші ресурси використовуються більш ефективно, що допомагає заощаджувати природні ресурси та гроші споживача.

- Керує завданнями. Розумні віртуальні помічники, такі як Google Home або Amazon Echo, можуть виконувати завдання за допомогою розпізнавання мови та голосових команд. Наприклад, власники будинків можуть використовувати голосові команди для ввімкнення музики, пошуку в Інтернеті та керування домашніми розумними пристроями.

Однак системам домашньої автоматизації важко стати мейнстрімом, частково через їхню технічну природу. Серед поширених недоліків "розумного" будинку можна виділити наступні:

- Потребує надійного підключення до інтернету. Ненадійне інтернет-з'єднання або перебої в роботі мережі можуть призвести до того, що пристрої та гаджети, підключені до розумного будинку, не працюватимуть.

- Уявна складність. Деяким людям важко або не вистачає терпіння працювати з технологіями. Виробники розумних будинків і альянси

працюють над зниженням складності і поліпшенням користувацького досвіду, щоб зробити його приємним і корисним для користувачів з будь-яким технічним рівнем.

- Відсутність стандартів. Щоб системи домашньої автоматизації були дійсно ефективними, пристрої повинні бути сумісними незалежно від виробника і використовувати один і той же протокол або, принаймні, взаємодоповнюючі протоколи. Оскільки це відносно новий ринок, золотого стандарту для домашньої автоматизації ще не існує. Однак стандартні альянси співпрацюють з виробниками та протоколами, щоб забезпечити сумісність та безперебійну роботу користувачів.

- Сумнівна безпека. Пристрої Інтернету речей створюють проблеми з безпекою, оскільки більшість з них не мають вбудованого шифрування. Крім того, вони можуть слугувати точками доступу до конфіденційних даних ширшої мережі, збільшуючи поле для атак. Згідно з нещодавнім звітом компанії Parks Associates, яка займається дослідженням споживчого ринку Інтернету речей, 55% споживачів стурбовані безпекою своїх пристроїв "розумного дому". Якщо хакери зможуть проникнути в розумний пристрій, вони потенційно можуть вимкнути світло, сигналізацію і відімкнути двері, залишивши будинок беззахисним перед зломом.

- Відсутність конфіденційності даних. Багато власників розумних будинків також турбуються про конфіденційність даних. Згідно з дослідницьким звітом Parks Associates, близько 72% споживачів висловили занепокоєння або сильне занепокоєння щодо безпеки їхніх персональних даних, зібраних і переданих пристроями розумного будинку. Так само вони стурбовані потенційним несанкціонованим доступом або контролем над розумними пристроями без їхнього дозволу. Хоча виробники пристроїв і платформ для "розумного дому" збирають дані про споживачів, щоб краще адаптувати свої продукти

або пропонувати клієнтам нові та вдосконалені послуги, довіра і прозорість є критично важливими для виробників, які прагнуть залучити нових клієнтів.

- Витрати. Незважаючи на те, що ціни знижуються, багато пристроїв "розумного дому" все ще залишаються дорогими, а переобладнання всього будинку може коштувати тисячі доларів.

Етапи розвитку розумного будинку

За останні кілька десятиліть технологія розумного будинку пройшла довгий шлях розвитку. Наведена нижче хронологія показує важливі події в історії технології розумного будинку:

- 1975. З виходом X10, комунікаційного протоколу для домашньої автоматизації, розумний дім, який колись був нездійсненною мрією а-ля "Джетсони", став реальністю. X10 надсилає радіосигнали радіочастотою 120 кГц з цифровою інформацією в існуючу електропроводку будинку до програмованих розеток або вимикачів. Ці сигнали передають команди відповідним пристроям, контролюючи, як і коли вони працюють. Наприклад, передавач може надсилати сигнал по електропроводці будинку, наказуючи пристрою увімкнутися в певний час. Однак, оскільки електрична проводка не призначена для захисту від радіочастотного шуму, X10 не завжди була повністю надійною. Сигнали губилися, а в деяких випадках вони не перетинали ланцюги, які були прокладені на різних полярностях, що виникало, коли 220-вольтова мережа розділялася на пару 100-вольтових ліній, як це часто буває в США. Крім того, X10 спочатку була односторонньою технологією, тому, хоча розумні пристрої могли приймати команди, вони не могли надсилати дані назад до центральної мережі. Пізніше, однак, з'явилися двосторонні пристрої X10, хоча і за вищою ціною.

- 1984. Американська асоціація будівельників придумала термін "розумний дім", щоб просувати концепцію технологій у дизайні житла.

- 2005. Компанія з домашньої автоматизації Insteon представила технологію, яка поєднала електричну проводку з бездротовими сигналами. З того часу з'явилися інші протоколи, зокрема Zigbee і Z-Wave, щоб протистояти проблемам, до яких схильний X10.

- 2007. З'явилися перші смарт-телевізори. Вони пропонували інтегровані послуги, підключені до Інтернету, такі як потокове мовлення та доступ до створеного користувачем контенту.

- 2011. Новостворена компанія Nest Labs випустила свій перший розумний продукт - термостат Nest Learning Thermostat. Компанія також створила розумні детектори диму та чадного газу і камери безпеки. Після придбання Google у 2015 році Nest стала дочірньою компанією Alphabet Inc.

- 2012. SmartThings Inc. розпочала кампанію на Kickstarter, зібравши 1,2 мільйона доларів для фінансування своєї системи розумного будинку. Після додаткового фінансування компанія вийшла на ринок у серпні 2013 року, а в 2014 році була придбана компанією Samsung.

- 2014. Представлено Amazon Echo, Amazon Alexa та Apple HomeKit, які зробили гігантський стрибок у сфері розумних пристроїв з голосовим управлінням.

- 2016-2018. Поява розумних колонок, таких як Google Home, Google Nest, Apple HomePod і Sonos, сигналізувала про значний зсув у тому, як користувачі взаємодіють з розумними домашніми пристроями.

- Сьогодні. Тенденції домашньої автоматизації продовжують розвиватися, з'являється все більше варіантів підключення та функцій [1].

1.2 Технічне завдання

Розумний будинок – це сучасна інтелектуальна система, яка об'єднує різні технології, а також пристрої, які беруть участь у функціонуванні системи з метою автоматизації та покращення комфорту, безпеки й ефективності життя у будинку. Такий будинок використовує різні сенсори, датчики, зв'язок та програмне забезпечення для моніторингу, а також керування різними аспектами повсякденного життя [2].

Система моніторингу безпеки розумного будинку включає в себе спеціалізовані сенсори, камери відеоспостереження, датчики руху й інші пристрої, які призначені для виявлення потенційних загроз та незвичайних подій у приміщенні та на території будинку. Система виявляє, аналізує та попереджує користувача про можливі ситуації, які вимагають його уваги та реагування на потенційну загрозу.

У процесі проєктування системи моніторингу безпеки розумного будинку важливо враховувати всі ключові характеристики об'єкта, типи сенсорів, а також програмне забезпечення для ефективного функціонування та забезпечення безпеки мешканців та майна. Саме тому під час проєктування важливо враховувати параметри, характеристики, які були попередньо описані у пункті 1.1.

У процесі розробки для системи моніторингу периметру безпеки розумного будинка необхідно забезпечити зберігання наступних даних:

- Інформація про користувачів:
    - Дані про користувачів системи, включаючи їх ім'я, електронну пошту та роль (адміністратор, користувач тощо).
    - Ідентифікатори користувачів та дані для авторизації (логін та пароль).
- Повідомлення користувачам:
    - Інформація про сформовані та надіслані повідомлення

користувачам, включаючи тип повідомлення, текст та час надсилання.

- Дані про датчики:
  - Інформація про датчики, розміщені в розумному будинку, включаючи назву, тип, місцезнаходження, дату встановлення та будинку, в якому вони розташовані.
- Інформація про будинок:
  - Дані про будинок, який є у власності користувача, включаючи адресу, тип будинку та користувача, який є власником будинку.
- Дані про лічильники:
  - Інформація про лічильники, встановлені в будинку, включаючи тип лічильника та будинок, до якого він відноситься.
- Відеозаписи:
  - Інформація про відеозаписи, зняті системою моніторингу безпеки, включаючи будинок, дату та час запису та шлях до файлу з відео.

Усі ці дані дозволяють розроблюваній системі моніторингу безпеки вести облік користувачів, повідомлень, датчиків, будинків, лічильників та відеозаписів, що є надзвичайно важливим для забезпечення безпеки та ефективного контролю у всіх процесах роботи розумного будинку. Вони можуть бути використані у процесі аналізу, а також подальшої оптимізації системи моніторингу та реагування на всі події, які відбуваються у будинку. Для ефективної роботи системи необхідно забезпечити внесення в оперативну БД вищезазначеної інформації, а також розробити структуру СД для подальшої передачі історичних даних у неї. Це дозволить краще проаналізувати процес моніторингу та подальших змін у внутрішній екосистемі.

Під час аналізу використані інструменти OLAP, а також Data Mining можуть надати відповідь на такий список можливих запитань:

1) Чи є залежність між показниками індикаторів та кількістю сповіщень, отриманих в певному районі, і як ця залежність змінюється з часом?
2) Чи є сезонні зміни в показниках індикаторів, і як це впливає на кількість сповіщень?
3) Чи є будинки, які найчастіше отримують сповіщення, і які типи індикаторів є найбільш активними для цих будинків?
4) Чи є залежність між показниками індикаторів та кількістю сповіщень, отриманих в певному районі, і як ця залежність змінюється з часом?

1.3 Аналіз наявних рішень

У процесі вивчення та аналізу предметної області було проведено аналіз існуючих робіт, які пов’язані з вирішенням питання моніторингу безпеки розумного будинку. Нижче наведено результати, отримані під час ознайомлення з різноманітними наявними рішеннями.

Автори Раджив Піяре та Сон Ро Лі у своїй роботі «Система керування та моніторингу розумного будинку за допомогою смартфона» описали недорогу та гнучку систему домашнього контролю та моніторингу з використанням вбудованого мікро-веб-сервера, з IP-з'єднанням для доступу та управління пристроями і приладами віддалено за допомогою додатку для смартфонів на базі Android. Запропонована система не потребує виділеного серверного комп'ютера і пропонує новий протокол зв'язку для моніторингу та управління домашнім середовищем, який має більше, ніж просто функцію перемикання функцією перемикання. Розроблений мобільний додаток для смартфона для домашнього контролю та моніторингу надає користувачеві наступні функції:

1) Віддалене підключення до домашнього шлюзу.
2) Керування пристроями.
3) Моніторинг пристроїв.
4) Керування розкладом.

На рис. 1.2 показано графічний інтерфейс користувача для контролю та управління домашнім середовищем за допомогою смартфона [3].

Рис. 1.2 Інтерфейс системи

Розроблена система дозволяє проводити моніторинг пристроїв та управління ними, проте не дозволяє проводити аналіз для подальшого підвищення ефективності роботи системи розумного будинку.

Іншим комерційним проєктом у галузі моніторингу безпеки розумного будинку є Monitoreal, який пропонує програмне забезпечення для моніторингу, яке оптимізоване для використання на смартфоні чи планшеті. Також вони розробили компактний і простий у використанні пристрій, який інтегрується у нову або існуючу систему камер спостереження [4].

Ключовими особливостями розробленої системи є:

- Пряма трансляція: перегляд відео в реальному часі з підключених камер безпеки прямо з додатку Monitoreal.

- Перегляди з декількох камер: перемикання між різними видами камер у різних місцях, щоб отримати цілісне уявлення про всі зони спостереження на одному екрані.

- Миттєві сповіщення: можливість отримувати миттєві сповіщення у разі будь-якої підозрілої активності.
- Релейні реакції: запуск підключених аксесуарів на основі заздалегідь визначених виявлень, щоб відлякати зловмисників ще до того, як вони проникнуть на територію, зокрема розумні світильники, сигналізації, сирени, стробоскопи, ворота, Bluetooth-колонки тощо.
- Моніторинг 24/7: отримання доступу до архівних відеозаписів, які надійно зберігаються на об'єкті, щоб у будь-який час переглядати або ділитися ними з іншими користувачами.
- Розумна аналітика: використання аналітичних інструментів Monitoreal на основі штучного інтелекту, які можуть розрізняти нормальну та підозрілу активність, зменшуючи кількість хибних тривог.
- Пульт дистанційного керування: віддалене встановлення та зйомка системи з-під охорони, керування кутами огляду камер та іншими пристроями розумного дому прямо з застосунку.
- Зручний інтерфейс: інтуїтивно зрозумілий інтерфейс дозволяє орієнтуватися у функціях програми будь-кому, навіть якщо він не розбирається в технологіях [5].

Також одним з комерційних проєктів є Alexa Smart Home – екосистема від Amazon, в основі якої лежить пристрій Echo і асистент Alexa. Echo – це розумна колонка, яка обробляє голосові команди за допомогою Alexa. Використання Echo не є обов'язковим, оскільки клієнти можуть використовувати лише мобільний додаток для управління своїм.

Серед можливостей цієї системи є:

- шаблони для дій, що повторюються;
- охоронець, який надсилає сповіщення про тривогу з пристрою Echo;

- здогадки дозволяють Alexa запропонувати або нагадати користувачеві щось зробити (наприклад, зачинити двері, коли користувач відсутній);
- друк документів за допомогою голосових команд;
- інформаційна панель, яка містить аналітику для покращення роботи використовуючи показники, які включають вторгнення, припинення та необроблені запити будинком [6].

На ринку популярним рішень є також Google Home – продукт від компанії Google. Це програмне забезпечення для налаштування, керування, автоматизації та керування тисячами сумісних пристроїв у будинку [7].

Серед можливостей даного програмно-апаратного забезпечення можна виділити наступні:

- голосовий помічник Google Assistant для виконання певних завдань;
- керування розумним будинком: Google Home сумісний із різноманітними пристроями розумного дому, дозволяючи керувати світлом, термостатами, замками, камерами тощо за допомогою голосових команд або програми Google Home;
- інтеграція всіх сервісів екосистеми Google (календарі, зустрічі тощо);
- персоналізовані рекомендації: Google Home може надавати персоналізовані рекомендації на основі уподобань і попередніх взаємодій;
- інтеграція зі сторонніми службами та програмами: Google Home може підключатися до різноманітних сторонніх служб і платформ розумного дому, щоб розширити свої можливості та сумісність;
- підтримка мов: Google Home підтримує кілька мов і акцентів, що робить його доступним для глобальної аудиторії [6,8,9].

Ще одним технологічним рішенням є Hubitat Mobile. Це мобільний додаток дозволяє користувачам віддалено отримувати доступ до своїх інформаційних панелей Hubitat для моніторингу та керування пристроями з будь-якого місця, де є підключення до Інтернету. Інформаційні панелі можна налаштувати для надання доступу до певних пристроїв і варіантів використання [10].

Основними можливостями такої існуючої системи є:

- контроль пристроїв: програма дозволяє керувати окремими розумними пристроями, такими як освітлення, вимикачі, термостати та замки, з вашого мобільного пристрою;
- створення інформаційної панелі: можна створювати налаштовані інформаційні панелі для групування та керування кількома пристроями в одному місці для швидкого та легкого доступу;
- моніторинг енергії: можливість надавати інформацію про енергоспоживання та тенденції використання;
- інтеграція пристроїв спостереження: залежно від сумісності, можна переглядати прямі трансляції з камер безпеки в програмі;
- інтеграція голосового помічника: деякі версії програми можуть інтегруватися з голосовими помічниками, такими як Amazon Alexa або Google Assistant, для голосового керування [11].

2 МОДЕЛЮВАННЯ СИСТЕМИ

На етапі проєктування дуже важливо визначити які вимоги поставлені для роботи системи, щоб наступний крок вже безпосередньо розробки проходив ефективно та результативно. Для глибшого та детальнішого аналізу предметної області та вимог до розроблюваної системи під час проєктування розроблюються діаграми, які дають можливість отримати цілісну картину необхідного функціоналу, а також майбутнього плану роботи.

2.1 Структурно-функціональне моделювання

2.1.1 Діаграма SADT. Техніка структурованого аналізу та проектування (SADT) була розроблена Дугласом Т. Россом у 1974 році. Це корисна методологія для системного планування, аналізу вимог і проектування системи. SADT не розвинувся з техніки проектування, а був розроблений шляхом дослідження проблем, пов’язаних із визначенням системних вимог. Зазвичай він не використовується для розробки програм, оскільки конструкції, необхідні для послідовності, вибору та взаємодії, не представлені в SADT. Тому SADT слід використовувати на етапах планування, аналізу та загального проектування [20].

Діяльність SADT описує декомпозицію діяльності. Дані включені в модель діяльності як входи, виходи, елементи керування та механізми. Потім діаграма верхнього рівня розкладається та представляється на окремих діаграмах. Усі дані, пов’язані з даною діяльністю, явно відображаються (зазвичай більш детально) на діаграмах нижчого рівня.

Прямокутник в центрі описується дієсловом або дією. Ліва частина поля використовується для показу вхідних даних (позначених іменником). Це дані, які мають бути перетворені в результаті діяльності. Права сторона вікна показує вихідні дані, які є даними, перетвореними діяльністю, представленою в полі, і мають використовуватися в іншому місці. Верхня частина вікна використовується для показу керуючих даних, тобто даних, які обмежують виконання дії. Ця інформація має дві основні мети:

1. Це дозволяє розробнику системи явно показувати дані, які не перетворюються на вихідні дані, а замість цього використовуються для зміни поведінки діяльності.

2. Це дозволяє дизайнеру оцінити зв’язність і функціональне представлення всіх блоків на діаграмі. Необхідно представити зв’язки обмежень, щоб розрізнити ступені зв’язування та дозволити розробнику виконати якісну оцінку декомпозиції.

Нижня частина прямокутника на діаграмі діяльності використовується для показу допоміжного механізму діяльності. Тобто, якщо розробнику системи потрібно описати організації, які виконують певну діяльність, ця стрілка використовується для визначення відділу, підрозділу або навіть особи, відповідальної за цю діяльність. Сторона механізму коробки також може використовуватися в моделях перехресних посилань. На рисунку 2.1 представлено SADT діяльності.

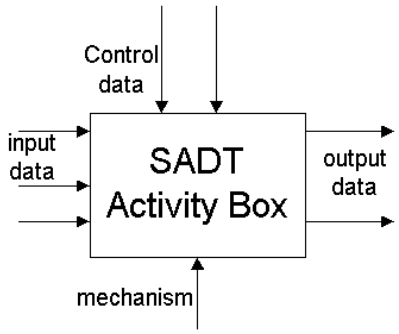

Рис. 2.1 Опис бізнес-процесу з використанням SADT

Переваги SADT:

- Забезпечує точний метод графічного представлення складних системних вимог програмного забезпечення.
- Забезпечує чіткий низхідний підхід до декомпозиції проблеми.
- Відокремлює дані від керування шляхом відокремлення цих функціональних областей у процесі діаграми.

- Сприяє командній роботі та погодженню специфікацій перед продовженням проектування.
- Вимагає, щоб рішення високого рівня приймалися на ранній стадії, забезпечуючи таким чином міцну основу для інших рішень.
- Дозволяє людям, які не займаються програмним забезпеченням, зрозуміти вимоги.
- Надає простий спосіб вимірювання прогресу шляхом вимірювання кількості діаграм, реалізованих у певний час.

Недоліки використання SADT:

- Може викликати труднощі, тому що інколи важко мислити лише функціональними термінами.
- Не підтримує певну методологію проектування.
- Потребує багато часу на підготовку.
- Діаграми може бути важко зрозуміти через додаткову керуючу інформацію в системах, де є багато діаграм, розташованих в ієрархічному порядку.
- Вимагає суворої дисципліни між розробниками, що ускладнює дизайнерам, які працюють незалежно, щоб показати потік продуктів між процесами [12].

Нижче на рис. 2.2 зображено SADT для системи моніторингу периметру безпеки розумного будинку.

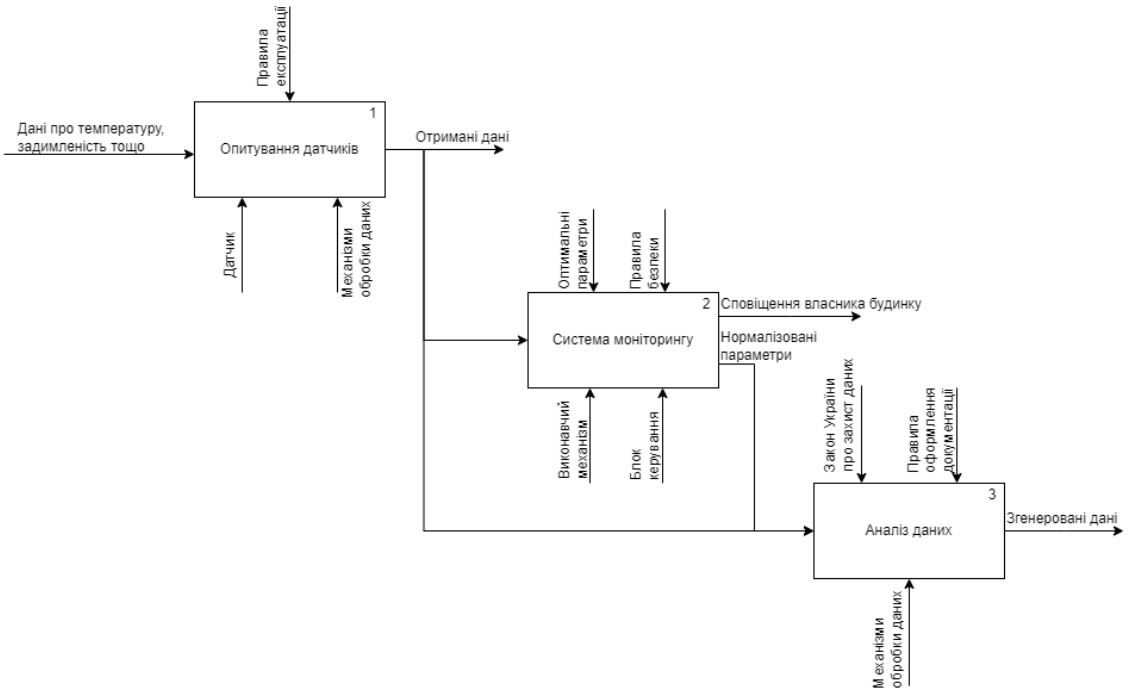

Рис. 2.2 SADT 1 рівня для системи моніторингу периметру безпеки розумного будинку

2.1.2 Діаграма DFD. Діаграма потоку даних (DFD) відображає потік інформації для будь-якого процесу чи системи. Він використовує визначені символи, такі як прямокутники, кола та стрілки, а також короткі текстові мітки, щоб показати вхідні дані, виходи, точки зберігання та маршрути між кожним пунктом призначення. Блок-схеми даних можуть варіюватися від простих, навіть намальованих від руки оглядів процесів, до поглиблених багаторівневих DFD, які все глибше вивчають, як обробляються дані. Їх можна використовувати для аналізу існуючої системи або моделювання нової. Як і всі найкращі діаграми та діаграми, DFD часто може візуально «сказати» речі, які було б важко пояснити словами, і вони працюють як для технічної, так і для нетехнічної аудиторії, від розробника до генерального директора. Ось чому DFD залишаються такими популярними після багатьох років. Хоча вони добре працюють для програмного забезпечення та систем потоку даних, сьогодні вони менш застосовні для візуалізації інтерактивного програмного забезпечення чи систем, що працюють у реальному часі або орієнтованих на бази даних.

У нотації діаграми DFD є чотири компоненти. Вони можуть мати різне позначення (рис. 2.3), проте мають однакове значення.

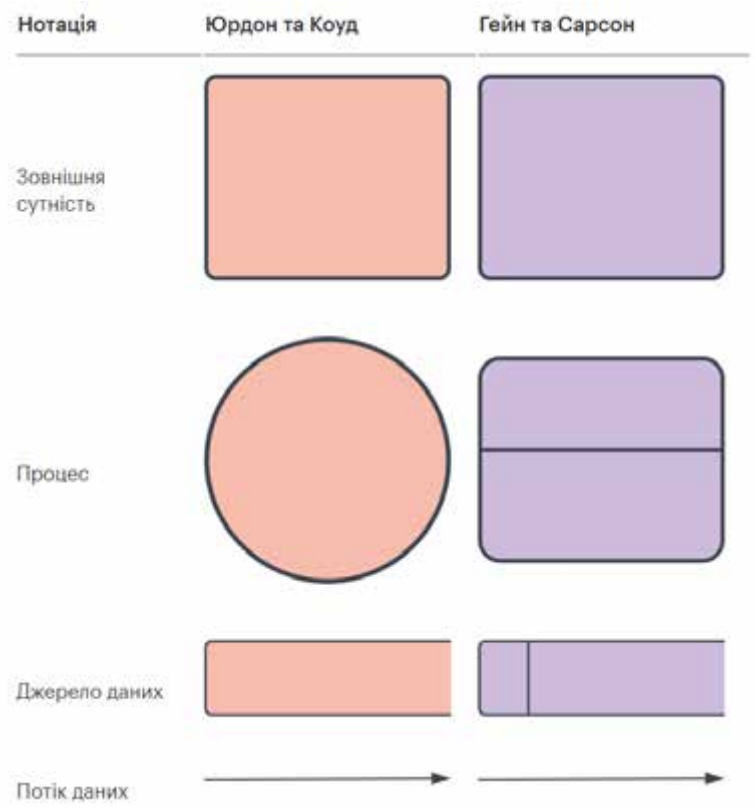

Рис. 2.3 Позначення на діаграмі DFD

Зовнішня сутність: зовнішня система, яка надсилає або отримує дані, спілкуючись із системою, на якій зображено схему. Вони є джерелами та адресатами інформації, що надходить або виходить із системи. Це може бути зовнішня організація чи особа, комп'ютерна система чи бізнес-система. Вони також відомі як термінатори, джерела та поглиначі або актори. Зазвичай вони малюються на краях діаграми.

Процес: будь-який процес, який змінює дані, створюючи результат. Він може виконувати обчислення, або сортувати дані на основі логіки, або керувати потоком даних на основі бізнес-правил. Для опису процесу використовується коротка мітка, наприклад «Надіслати платіж».

Сховище даних: файли або сховища, які містять інформацію для подальшого використання, наприклад таблицю бази даних або форму членства. Кожне сховище даних отримує просту мітку, наприклад «Замовлення».

Потік даних: маршрут, яким дані проходять між зовнішніми об’єктами, процесами та сховищами даних. Він відображає інтерфейс між іншими компонентами та відображається стрілками, зазвичай позначеними короткою назвою даних, як-от «Платіжна інформація» [13].

На рис. 2.4 зображено DFD для системи моніторингу. Для побудови була використана нотація Гейна-Сарсона. На діаграмі ми можемо побачити, які зовнішні сутності беруть участь в управлінні тими чи іншими процесами, які дані вони надають до конкретного процесу та до якого сховища даних направляються вихідні дані із заданого бізнес-процесу.

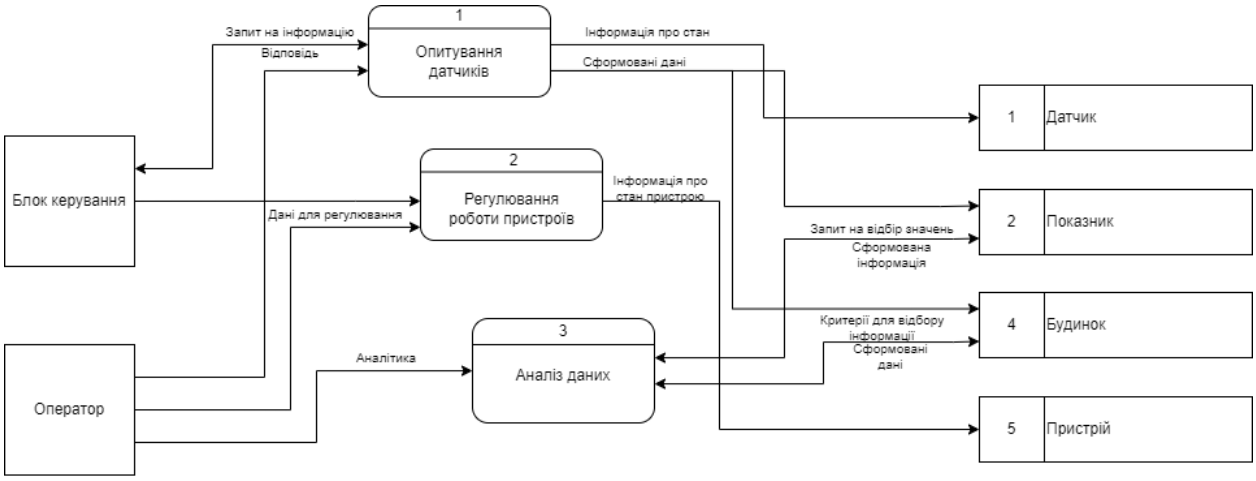

Рис. 2.4 DFD діаграма для системи моніторингу периметру безпеки розумного будинку

2.2 Об’єктно-орієнтоване моделювання

Об'єктно-орієнтована (ОО) парадигма — це стратегія розвитку, заснована на концепції, згідно з якою системи повинні будуватися з набору повторно використовуваних компонентів, які називаються об'єктами. Замість розділення

даних і функцій, як це робиться в структурованій парадигмі, об’єкти охоплюють і те, і інше [14].

UML, скорочення від Unified Modeling Language, — це стандартизована мова моделювання, що складається з інтегрованого набору діаграм, розроблених для допомоги розробникам систем і програмного забезпечення у визначенні, візуалізації, конструюванні та документуванні артефактів програмних систем, а також для бізнес-моделювання та інші непрограмні системи. UML представляє збірку найкращих інженерних практик, які виявилися успішними в моделюванні великих і складних систем. UML є дуже важливою частиною розробки об’єктно-орієнтованого програмного забезпечення та процесу розробки програмного забезпечення. UML використовує переважно графічні нотації для вираження дизайну проектів програмного забезпечення [15].

2.2.1 Діаграма прецедентів. В уніфікованій мові моделювання (UML) діаграма варіантів використання може узагальнити деталі користувачів системи (також відомих як актори) та їх взаємодію з системою.

Основними компонентами діаграми прецедентів є:

- Актори: користувачі, які взаємодіють із системою. Актором може бути особа, організація або зовнішня система, яка взаємодіє з вашою програмою чи системою. Вони мають бути зовнішніми об’єктами, які виробляють або споживають дані.

- Система: певна послідовність дій і взаємодій між акторами та системою. Систему також можна назвати сценарієм.

- Цілі: кінцевий результат більшості випадків використання. Успішна діаграма повинна описувати дії та варіанти, використані для досягнення мети.

Символи та позначення на діаграмі випадків використання

Нотація для діаграми варіантів використання досить проста і не містить стільки типів символів, як інші діаграми UML. Основними елементами діаграми прецедентів є:

- прецедент: овали горизонтальної форми, які представляють різні способи використання, які можуть мати користувачі;
- актори: фігурки, які представляють людей чи об'єкти, які взаємодіють із прецедентом;
- асоціації: межа між акторами та варіантами використання. У складних діаграмах важливо знати, які актори пов’язані з якими варіантами використання;
- системні граничні рамки: рамки, які встановлюють область використання системи для випадків використання. Усі нестандартні випадки використання розглядатимуться поза межами цієї системи;
- пакети: форма UML, яка дозволяє об’єднувати різні елементи в групи. Подібно до діаграм компонентів, ці групи представлені у вигляді папок файлів [16].

При побудові діаграми прецедентів важливо визначити перелік акторів, які взаємодіють з системою. На таблиці 1 наведений перелік акторів, які взаємодіють із системою.

Таблиця 1

Актори, які взаємодіють із системою

Актор	Опис
1	**2**
Оператор	Спостерігає за показниками, які вимірюються у розумному будинку, якщо необхідно, регулює коректність роботи підключених пристроїв, опрацьовує отримані значення з датчиків, вносить інформацію про будинок тощо
Аналітик	Проводить аналіз роботи, а також формує результуючу аналітичну звітність
Власник будинку	Приймає рішення щодо керування розумним будинком
Блок керування	Зчитує дані з підключених датчиків та керує роботою всіх необхідних виконавчих механізмів
Датчики	Проводить вимірювання певних показників на території розумного будинку

Проаналізувавши, які актори діють у системі, можна перейти до аналізу та опису прецедентів, які описують функціонал розроблюваної системи (таблиця 2). На основі обов’язків акторів та прецедентів була побудована діаграма прецедентів, яка зображена на рис. 2.5.

Дивлячись на те, що мета системи зосереджена на моніторингу даних, а також їхній подальший аналіз, тому головними акторами тут виступають блок керування та оператор. Саме вони відповідають за коректну роботу системи розумного будинку, а вже аналітик за проведення аналізу всіх отриманих показників. На основі отриманої аналітичної звітності власник будинку застосовує необхідні зміни у необхідних процесах функціонування розумного будинку. Також у системі акторами виступають датчики. Вони виконують

допоміжну функцію моніторингу у будинку, проте від їхньої коректності роботи залежить функціонування будинку загалом.

Таблиця 2

Перелік прецедентів у системі

Актор	Найменування	Формулювання
1	**2**	**3**
Датчики	Вимірювання параметрів у розумному будинку	Моніторинг показників у розумному будинку
Виконавчий механізм	Регулювання процесів роботи у розумному будинку	Здійснення необхідних дій для коректного функціонування розумного будинку
Блок керування	Зчитування даних з датчиків	Зчитує дані з підключених датчиків
Блок керування	Управління роботою виконавчих механізмів	Можливість керування підключених виконавчих механізмів
Оператор	Формування щоденних звітів	Можливість створення звітів використовуючи значення за вказаними критеріями
Оператор	Спостереження за роботою виконавчих механізмів	Спостерігає за станом та роботою системи моніторингу
Оператор	Ручне управління пристроями	Можливість зміни стану роботи виконавчих механізмів у ручному режимі

Таблиця 2 (завершення)

1	2	3
Аналітик	Формування аналітичних запитів	Створення запитів для проведення аналізу даних
Аналітик	Обчислення показника ефективності (KPI)	Проведення обчислень ключового показника ефективності
Аналітик	Формування аналітичної звітності	Можливість створення звітів на основі результатів проведеного аналізу даних
Аналітик	Перегляд аналітичної звітності	Опрацювання розроблених звітів з попередньо проаналізованими даними
Власник будинку	Прийняття рішень щодо роботи розумного будинку	Процес прийняття необхідних рішень у роботі розумного будинку на основі отриманих результатів аналізу

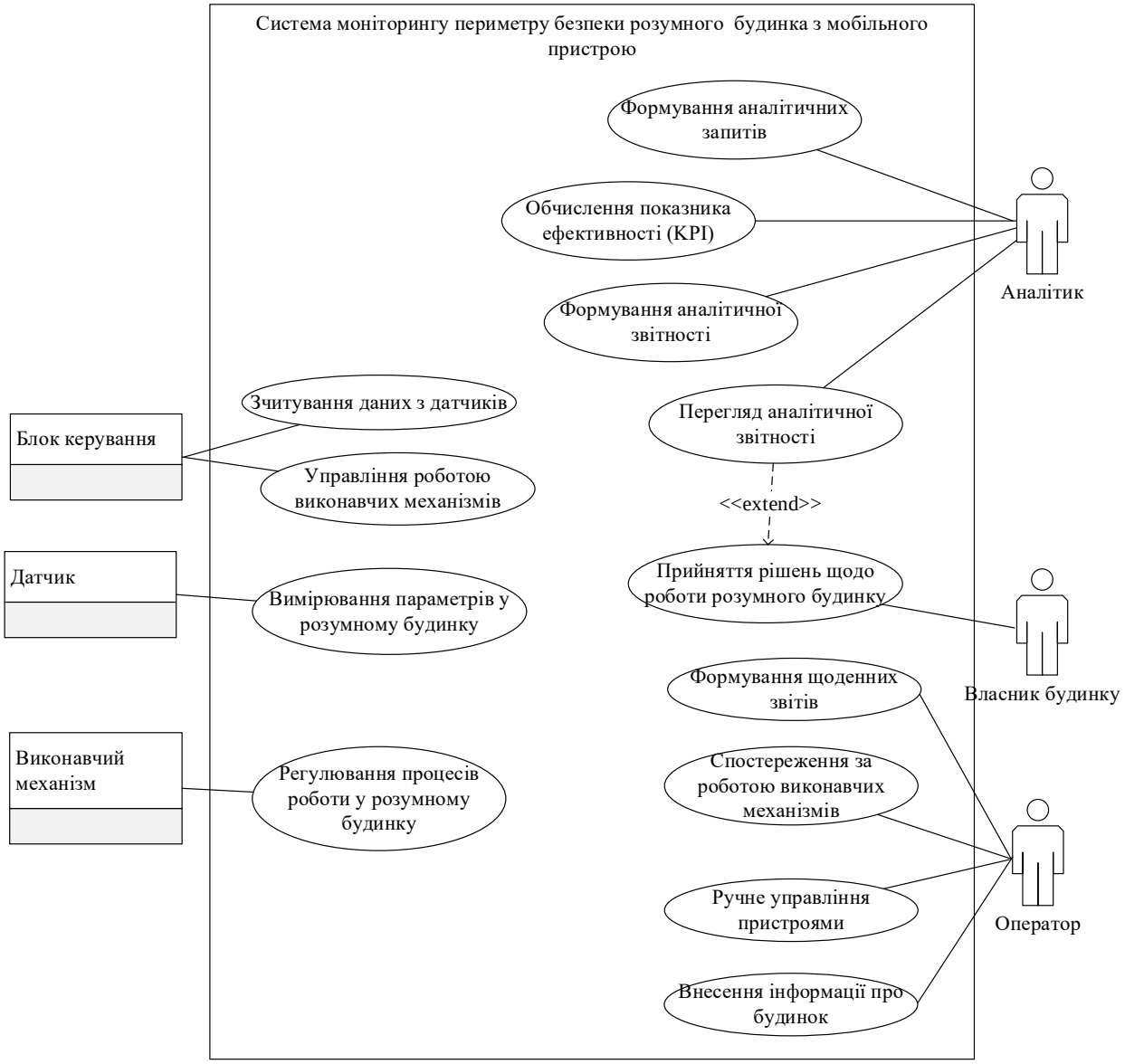

Рис. 2.5 Діаграма прецедентів системи моніторингу периметру безпеки розумного будинку

2.2.2 Діаграма діяльності. Діаграми діяльності описують, як діяльність координується для надання послуги, яка може бути на різних рівнях абстракції. Як правило, подія повинна бути досягнута деякими операціями, особливо якщо операція призначена для досягнення кількох різних речей, які вимагають координації, або як події в одному варіанті використання співвідносяться одна з одною, зокрема випадки використання, де дії можуть збігатися і вимагати узгодження. Він також підходить для моделювання того, як набір варіантів використання координується для представлення бізнес-процесів [17].

Основні компоненти діаграми діяльності:

- дія: етап діяльності, на якому користувачі або програмне забезпечення виконують певне завдання;
- вузол рішення: умовна гілка в потоці, представлена ромбом. Він включає один вхід і два або більше виходів;
- потоки керування: інша назва з’єднувачів, які показують потік між кроками на схемі;
- початковий вузол: символізує початок діяльності. Початковий вузол представлений чорним колом;
- кінцевий вузол: представляє останній крок у дії. Кінцевий вузол представлений окресленим чорним колом [18].

На рис. 2.6 продемонстрована діаграма діяльності системи. На діаграмі видно, що система має систему моніторингу показників, робота з виміряними даними.

На основі збережених даних, які показують процеси зміни певних показників у розрізі часу, відбувається аналіз інформації аналітиком та надання отриманої інформації до власника будинку. Від результатів аналізу залежить подальше прийняття необхідних рішень для роботи і проходження всіх процесів у розумному будинку.

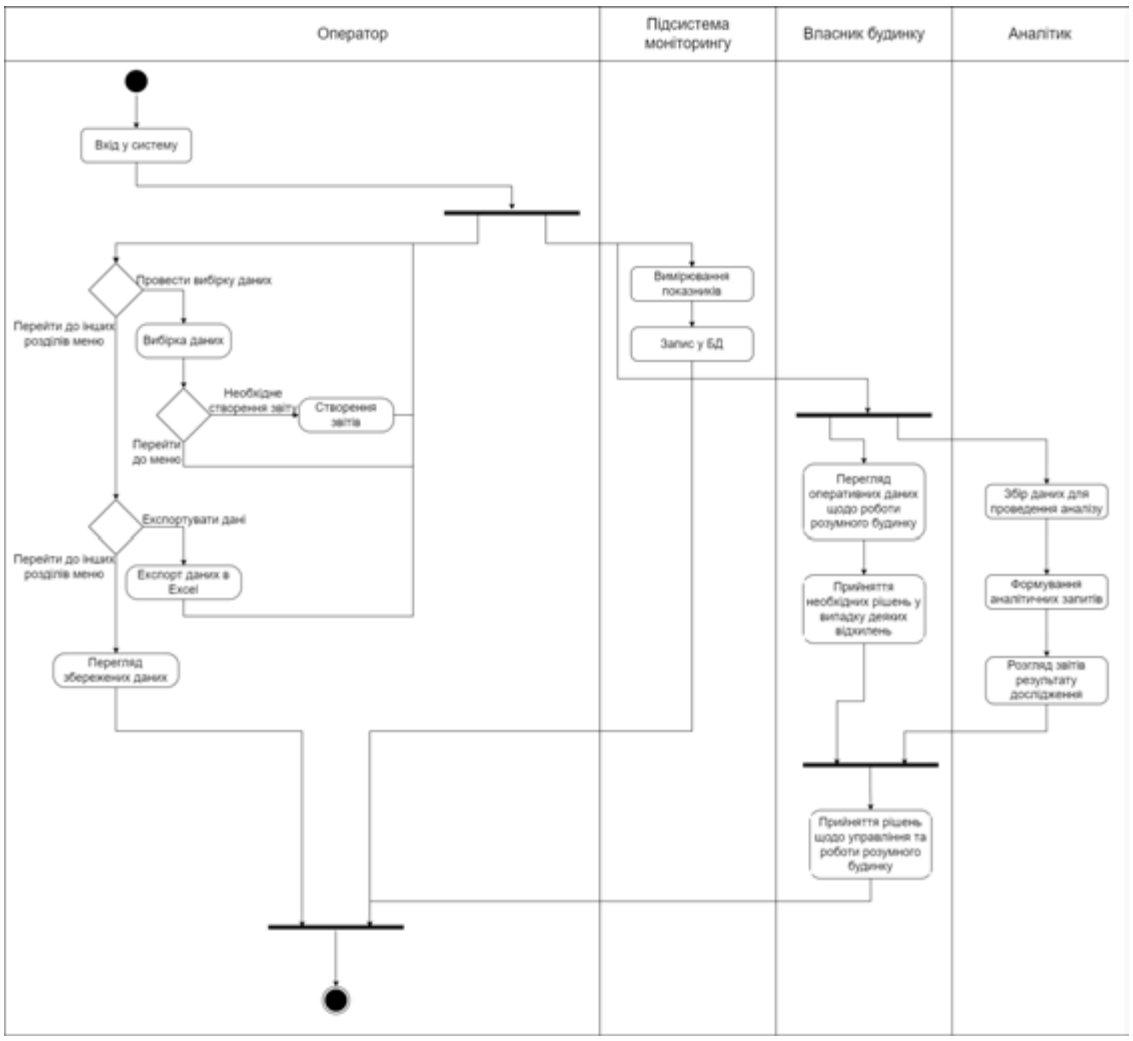

Рис. 2.6 Діаграма діяльності системи моніторингу периметру безпеки розумного будинку

2.2.3 Діаграма послідовності. Діаграма послідовності – це тип діаграми взаємодії, оскільки вона описує, як і в якому порядку група об’єктів працює разом. Ці діаграми використовуються розробниками програмного забезпечення та бізнес-професіоналами, щоб зрозуміти вимоги до нової системи або задокументувати існуючий процес. Діаграми послідовності іноді називають діаграмами подій або сценаріями подій.

Діаграми послідовності можуть бути корисними, щоб:

- детальніше представити діаграму прецедентів;
- моделювання логіки складної процедури, функції чи операції;
- деталізація того як об’єкти та компоненти взаємодіють один з одним,

щоб завершити певний процес;

- планування та розуміння детальної функціональності існуючого або майбутнього сценарію у роботі системи [19].

На рис. 2.7 наведено діаграму послідовності для проведення процесу аналізу даних. На основі інформації, яка експортована з системи моніторингу, відбувається накопичення даних у СД. Після цього аналітик формує необхідні запити для аналізу фактів і формує результуючу звітну інформацію, яка передається власнику будинку.

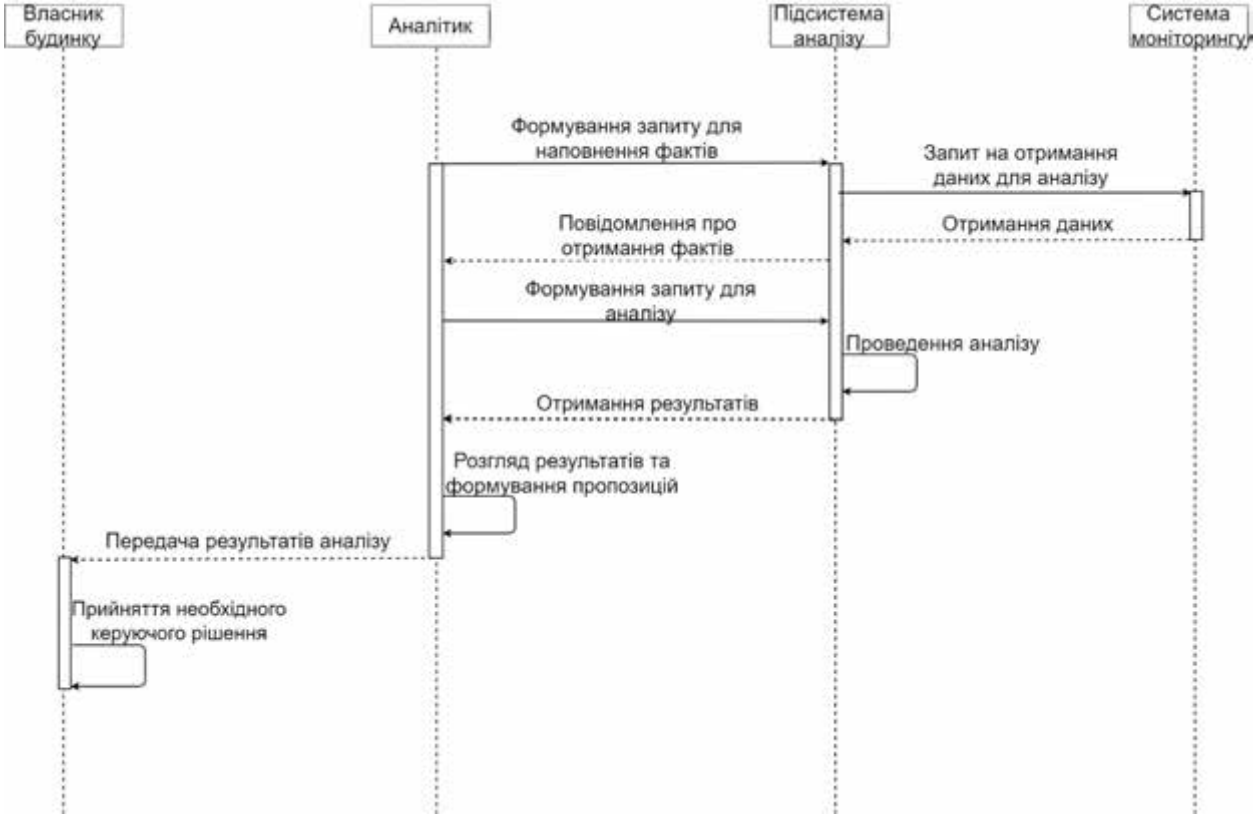

Рис. 2.7 Діаграма послідовності

3 РОЗРОБКА СИСТЕМИ

3.1 Інформаційне забезпечення системи

3.1.1 Логічна модель даних. Однією з найважливіших частин програмного забезпечення є інформація, над якою виконуються різноманітні операції. Створити ефективну структуру бази даних відразу досить складно, тому для візуалізації загальних понять та обов'язкових даних створюється логічна модель даних. Вона схематично демонструє ті інформаційні аспекти, що беруть участь в взаємодії системи та користувача.

Логічна модель даних системи моніторингу периметру безпеки розумного будинка з мобільного пристрою представлена на рисунку 3.1.

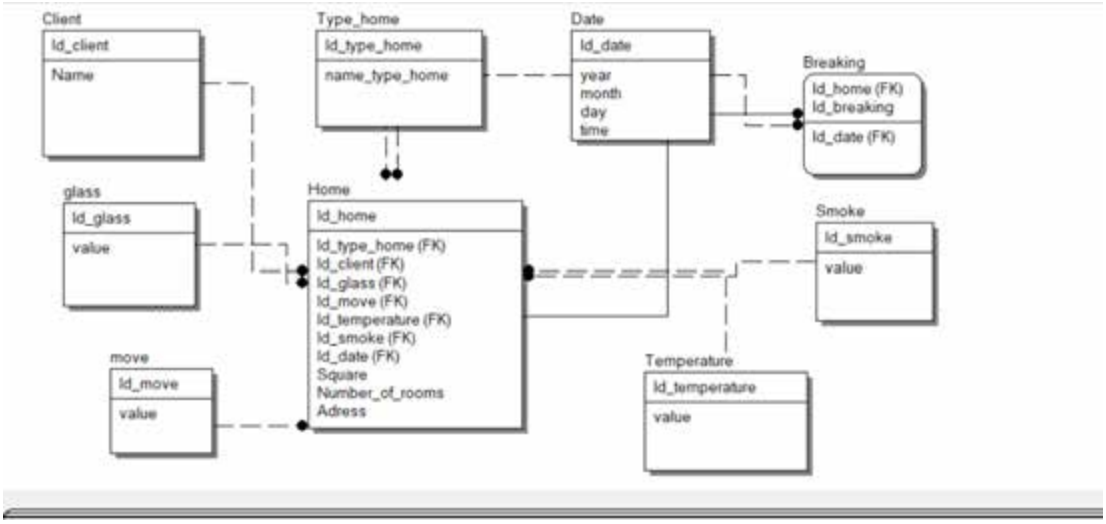

Рис. 3.1 Логічна схема даних

Загалом для системи моніторингу було виділено 9 основних сутностей:

- Client – сутність містить інформацію про клієнта: ключовий атрибут код клієнта(id_client) та ім'я клієнта (name). Основне призначення: зберігати дані користувачів, для авторизації їх в системі та ідентифікувати власника дому.
- Type_home – сутність містить інформацію про тип будинку: ключовий атрибут код типу будинку(id_type_home) та назву типу

будинку(name_type_home). Основне призначення: типізувати сутність Home.

- Date – сутність містить інформацію про дату: ключовий атрибут код дати(id_date) та 4 неключові, а саме рік, місяць, день та час. Основне призначення: зберігання часових показників системи.
- Breaking – сутність містить інформацію про крадіжки: ключовий атрибут коду крадіжки(id_breaking) та зовнішні атрибути код будинку й код дати. Основне призначення: зберігати дані про проникнення.
- Glass – сутність містить інформацію з датчику відкриття вікон: ключовий атрибут код датчику(id_glass) та значення датчику(value). Основне призначення: зберігати дані, отримані від датчика відкриття вікон.
- Move – сутність містить інформацію з датчику руху: ключовий атрибут код датчику(id_muve) та значення датчику(value). Основне призначення: зберігати дані, отримані від датчика руху.
- Temperature – сутність містить інформацію з датчику температуру: ключовий атрибут код датчику(id_temperature) та значення датчику(value). Основне призначення: зберігати дані, отримані від датчика температури.
- Smoke – сутність містить інформацію з датчику диму: ключовий атрибут код датчику(id_smoke) та значення датчику(value). Основне призначення: зберігати дані, отримані від датчика диму.
- Home – сутність містить інформацію про будинок: містить ключовий атрибут(id_home) та зовнішні ключі з всіх інших таблиць, а також 3 неключові, а саме: площу будинку(square), кількість кімнат(Number_of_rooms), та адресу(Adress).Основне призначення: зберігати дані про будинок користувача та ідентифікувати датчики.

3.1.2 Середовище проектування БД. Realm – це новий модуль бази даних, який покращує спосіб використання баз даних, а також підтримує зв'язки між об'єктами [21].

Realm – це об'єктна та мобільна база даних, що виступає рішенням для зберігання даних мобільних та веб-розробок. Більш простим визначенням буде те, що Realm – це місце, де зберігаються та керуються дані програми чи веб-сайту. Можна сказати, що мобільна база даних Realm є ідеальною заміною SQLite та Core Data. Realm пропонує рішення для баз даних в проектах, де застосовуються такі мови розробки Java, Kotlin, Swift, Obj-C, JavaScript і .NET.

Платформа Realm складається з двох основних компонентів:

- База даних Realm;
- Сервер Realm Object.

Ці два компоненти працюють у поєднанні для автоматичної синхронізації даних, надаючи великий спектр можливостей, починаючи від офлайн-додатків і закінчуючи складною серверною інтеграцією.

Realm допомагає розробникам грамотно працювати з непередбачуваним середовищем мобільних додатків, у якому пристрої можуть вимкнутись у будь-який час; з'єднання можуть бути втрачені тощо. Це надає можливість координації між мобільними клієнтами, серверними API та базами даних.

Особливості Realm полягає в:

- Локальне сховище Realm: Realm Database працює на клієнтських пристроях. Зберігати, отримувати доступ і оновлювати дані є простими та легкими, оскільки доступ до об'єктів здійснюється за допомогою рідної мови запитів для кожної платформи.
- Realm пропонує надійність мережі: База даних Realm насамперед офлайн. Це означає, що ви завжди читаєте та записуєте до локальної бази даних, а не через мережу. Коли Realm Sync увімкнено, Realm Database синхронізує дані з MongoDB Realm через мережу у фоновому потоці. Цей протокол синхронізації послідовно вирішує конфлікти на

кожному клієнті та у зв’язаному кластері MongoDB Atlas.

- Реактивний інтерфейс користувача: живі об’єкти завжди відображають найновіші дані, що зберігаються в Realm Database, і дозволяють підписатися на зміни, щоб ви могли постійно оновлювати свій інтерфейс користувача. Realm SDK дає змогу підключатися до локальних сфер для всіх різних мов розробки та платформ, таких як Android, iOS, Node.js, React Native і розробка UWP.

Переваги використання Realm:

- швидка та проста розробка;
- створення масштабних програм;
- зменшення затримки за рахунок сервера;
- можливість роботи офлайн;
- програма для кількох платформ додатків.

3.1.3 Реалізація фізичної моделі даних. На основі розробленої логічної моделі даних та визначення типу БД та середовище створюється оперативна база даних. Вона має відповідати всі вимогам системи та надавати зручний та швидкий спосіб взаємодії з інформацією. На рисунку 3.2 представлено фізичну модель даних системи моніторингу.

Загалом БД має 6 основних сутностей, 7 довідникових сутностей та 4 функціональних сутностей. Розглянемо основні сутності:

- User – сутність містить дані користувача, що має акаунт в системі. Вона містить такі поля:
  - user_id – ключове поле, що є ідентифікатором акаунта користувача;

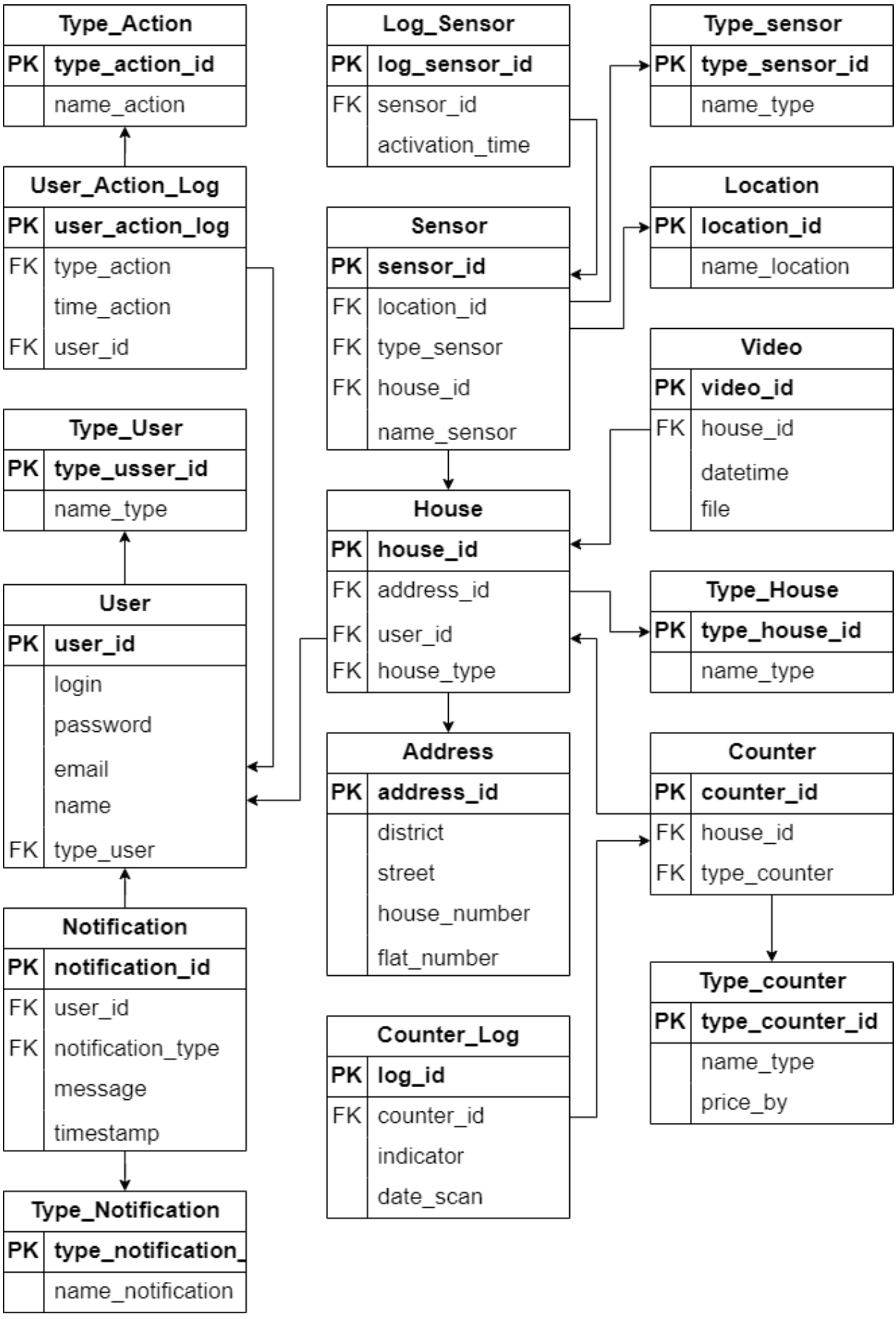

Рис. 3.2 Схема оперативної бази даних

- login та password – поля, що містять дані, призначені для авторизації в системі користувача;
- email – поле зберігає електронну адресу користувача, куди надходять найбільш важливі повідомлення;
- name – поле містить дані про ім’я користувача;
- type_user – поле представляє собою зовнішній ключ іншої сутності – Type_use, що визначає тип акаунта користувача, так як система має кілька ролів користувачів.

- Notification – сутність містить дані про сформовані та надіслані повідомлення на акаунт користувача. Вона має такі поля:

  - notification_id – ключове поле, що ідентифікує записи в сутності;
  - user_id – поле представляє собою зовнішній ключ іншої сутності – User, що визначає акаунт користувача, котрому було сформовано та надіслано повідомлення;
  - notification_type – поле представляє собою зовнішній ключ іншої сутності – Type_Notification, що визначає тип повідомлення;
  - message – поле містить сформований текст повідомлення;
  - timestamp – поле зберігає час надсилання повідомлення користувачу.

- Sensor – сутність призначена зберігати дані про датчики, що знаходяться в будинку користувача. Вона має такі поля:

  - sensor_id – ключове поле, що ідентифікує кожен датчик в системі;
  - location_id – поле представляє собою зовнішній ключ іншої сутності – Location, що визначає місцезнаходження датчика в будинку;
  - type_sensor – поле представляє собою зовнішній ключ іншої сутності – Type_Sensor, що визначає тип датчика, розміщеного саме в будинку користувача;
  - house_id – поле представляє собою зовнішній ключ іншої сутності

– House, що визначає в якому саме будинку розміщений датчик;

- name_sensor – поле зберігає назву датчика, що дозволяє користувачу більш точніше його ідентифікувати.

- House – сутність містить інформацію про будинок, що належить користувачу. Вона має такі поля:

  - house_id – ключове поле, що ідентифікує будинок, яким керує користувач;
  - address_id – поле представляє собою зовнішній ключ іншої сутності – Address, що містить адресу будинку;
  - user_id – поле представляє собою зовнішній ключ іншої сутності – User, що визначає власника будинку в системі;
  - house_type – поле представляє собою зовнішній ключ іншої сутності – Type_House, шо визначає тип будинку.

- Counter – сутність призначена зберігати дані про лічильник в будинку. Вона має такі поля:

  - counter_id – ключове поле, що ідентифікує лічильник;
  - house_id – поле представляє собою зовнішній ключ іншої сутності – House, що визначає належність лічильника до будинку;
  - type_counter – поле представляє собою зовнішній ключ іншої сутності – Type_counter, що визначає тип лічильника.

- Video – сутність містить дані про відео, що було знято з різних причин. Вона має такі поля:

  - video_id – ключове поле, що ідентифікує відеозапис;
  - house_id – поле представляє собою зовнішній ключ іншої сутності – House, що визначає в якому будинку було знято запис;
  - datetime – поле зберігає дату та час відеофіксації;
  - file – поле зберігає шлях до файлу з відео.

Сутності, що можна назвати довідниковими, призначені класифікувати інформацію, що дає можливість робити необхідні вибірки просто та досить швидко їх аналізувати. До таких сутностей в оперативній базі даних відносяться:

- Type_Action – сутність містить тип дії, яку користувач може виконати в системі.
- Type_User – сутність зберігає можливі типи акаунтів користувачів.
- Type_Notification – сутність містить інформацію про можливий тип повідомлень, що можуть надсилатись користувачу.
- Type_Sensor – сутність призначена зберігати дані про можливі типи датчиків, що можуть бути встановлені в будинку.
- Location – сутність містить назви кімнат, де можуть біти розташовані датчики в будинку чи ззовні.
- Type_House – сутність містить типи будинків, що присутні в системі.
- Type_Counter – сутність зберігає типи лічильників, присутніх в системі. Особливість даної сутності полягає в наявності додаткового поля – price_by, що зберігає тариф за використання ресурсу, який рахує лічильник.

До функціональних сутностей можна віднести ті таблиці, що необхідні саме системі, наприклад щоб не породжувати масиви схожих даних, що відрізняються лише кількома параметрами, а також, найчастіше, є наслідком нормалізації даних в реляційних БД. До таких сутностей можна віднести:

- User_Action_Log – сутність має зберігати дані про дії користувача в системі. Вона має поля, що пов’язують тип дії, користувача та час виконання дії.
- Log_Sensor – сутність зберігає дані про показники датчиків в певний момент часу.

- Counter_Log – сутність містить показники лічильників в певний день.
- Address – сутність зберігає адреси, за якими знаходяться будинки користувачів.

3.1.4 Проектування сховища даних. Сховище даних СД – предметно-орієнтований, інтегрований, незмінний, що підтримує хронологію, набір даних, організований для цілей підтримки прийняття рішень. Основна мета створення сховища в тому, щоб зробити усі значимі для управління бізнесом дані доступними в стандартизованій формі, придатними для аналізу та отримання необхідних звітів. Для досягнення цього потрібно отримати дані із існуючих внутрішніх та зовнішніх джерел [22].

Загалом проектування сховища даних потребує чіткого розуміння термінів:

- Вимір – це множина однотипних даних, що утворюють одну з граней куба і характеризують якусь ознаку показників, котрі знаходяться в комірці багатовимірного куба.
- Таблиця фактів – це центральна таблиця моделі типу зірки чи сніжинка, яка містить всі первинні ключі таблиць вимірів, а також значення показників, що аналізуються. Таблиця фактів зв’язується з таблицями вимірів і виступає підпорядкованою, дочірньою таблицею.
- Таблиці вимірів – це таблиці, що характеризують певні факти центральної таблиці фактів. Таблиці вимірів містять первинні ключі, за допомогою яких виконується зв’язок з таблицею фактів та характеристики ключових полів, як правило, це довідкові текстові дані, наприклад, це можуть бути дані про назву товару, назву його виробника, тип товару та інші. Таблиці вимірів виступають як батьківські по відношенню до таблиць фактів [23].

На рисунку 3.3 представлене сховище даних системи моніторингу периметру безпеки розумного будинка з мобільного пристрою.

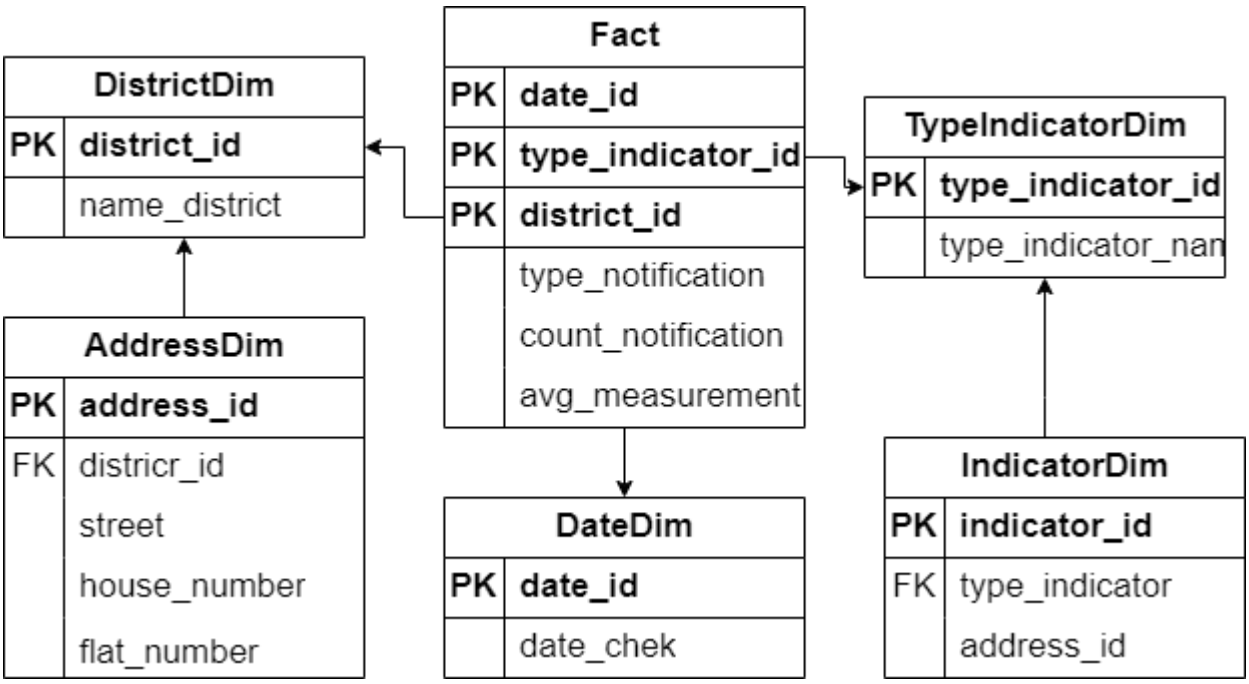

Рис. 3.3 Сховище даних

Сховище даних має таку структуру:

- DistrictDim – таблиця вимір, що дозволяє аналізувати дані в розрізі області, в якому знаходиться будинок користувача.
- AddressDim – таблиця вимір, що містить адресу будинків користувачів.
- DateDim – таблиця вимір, що дозволяє аналізувати дані в часовому розрізі.
- TypeIndicatorDim – таблиця вимір, що містить типи індикаторів системи.
- IndicatorDim – таблиця вимір, що зберігає дані про індикатор.
- Fact – таблиця фактів, що містить показники, що аналізуються та певним представленням сутності Notification оперативної бази даних.

3.2 Програмне забезпечення системи

3.2.1 Архітектура програмного забезпечення. Незважаючи на те, що система має бути реалізована як мобільний додаток, для забезпечення всіх поставлених вимог моніторингу периметру безпеки розумного будинка з мобільного пристрою, було обрано звичний тип архітектури – клієнт-серверну.

Основна концепція роботи клієнт-серверної архітектури полягає саме в зосереджені головних обчислювальних ресурсів саме на сервері. Основними складовими такої архітектури є:

- Сервер – вузол, що отримує, обробляє та повертає запити, що йому надсилає клієнт;
- Клієнт – вузол, що надсилає та отримує інформацію від сервера;
- Мережа – вузол забезпечує взаємодію сервера та клієнта.

Триланкова клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка застосунку. Програми проміжного рівня можуть функціювати під управлінням спеціальних серверів застосунків, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних[24].

Триланкова архітектура складніша, але завдяки тому, що функції розподілені між серверами другого і третього рівня, ця архітектура проявляє:

- високий ступінь гнучкості і масштабованості;
- високу безпеку (тому що захист можна визначити для кожного сервісу або рівня);
- високу продуктивність (тому що завдання розподілені між серверами).

На рисунку 3.4 представлена топологія системи моніторингу периметру безпеки розумного будинка з мобільного пристрою.

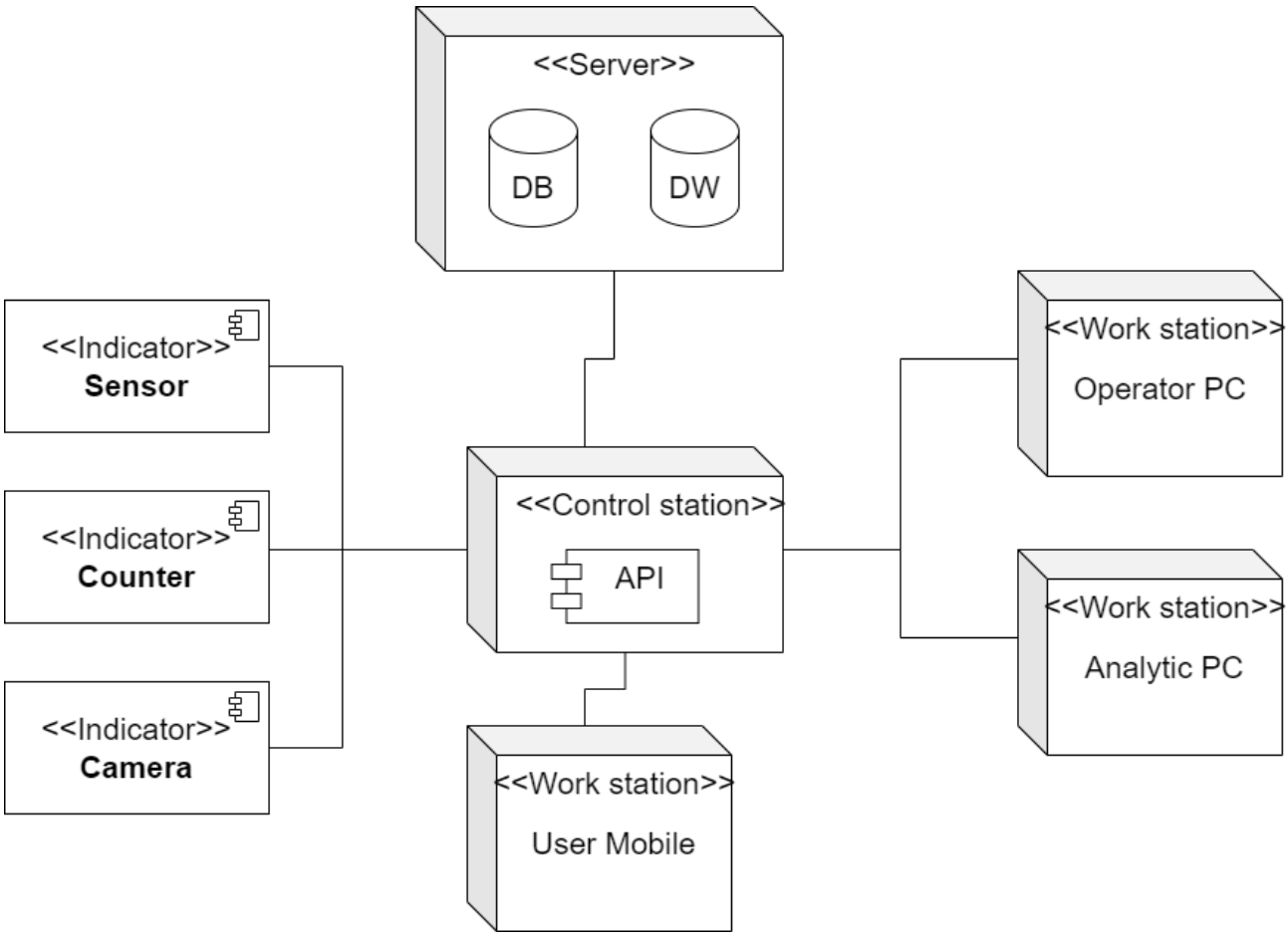

Рис. 3.4 Топологія системи

Розглянемо основні вузли представленої топології:

- Server – вузол містить базу даних та сховище даних;
- Work station – вузол представляє собою різні пристрої, через які взаємодіють різні типи користувачів системи;
- Control station – вузол представляє собою сервер, що має інтерфейси, призначені для роботи з іншими вузлами;
- Indicator – вузол представляє собою загальне поняття, що може бути представлене як лічильник, датчик руху чи щось схоже.

3.2.2 Обрані інструменти та програмні засоби для реалізації програмного забезпечення системи.

Java – це об’єктно-орієнтована мова програмування на основі класів, що була створена для підтримки незначної кількості залежностей, щоб скомпільований код Java міг працювати на всіх платформах без необхідності повторної компіляції [25].

Java має багато функцій, які роблять її популярним вибором серед розробників як веб-, так і мобільних програм. Її особливості наведені нижче:

- Об’єктно-орієнтований: так само, як Python, C++ і Ruby, Java є об’єктно-орієнтованою мовою програмування. Усе є «об’єктом», який містить код і дані.

- Кросплатформність: оскільки програми Java спочатку перетворюються на байт-код компілятором, код може працювати на будь-якій машині, яка підтримує середовище виконання Java, що робить його незалежним від платформи.

- Безпека: оскільки Java не залежить від платформи та майже не взаємодіє з операційною системою, це робить мову набагато безпечнішою, ніж інші мови програмування.

- Висока продуктивність: незважаючи на те, що Java є інтерпретованою мовою, на відміну від C або C++, які скомпільовані, вона має високу продуктивність завдяки своєму власному своєчасному компілятору.

- Багатопотоковий: за допомогою Java ви можете писати програми , які виконують кілька завдань в окремих потоках. Наприклад, програма Java може надавати користувачам форму входу, одночасно запускаючи фонові процеси.

- Відкритий вихідний код: протягом багатьох років Java накопичила велику колекцію бібліотек з відкритим вихідним кодом, які значно полегшують розробку мобільних програм Java.

- Підтримка спільноти: Оскільки Java є старішою мовою програмування,

вона має велику спільноту розробників, які діляться цінними ідеями та знаннями, коли йдеться про розробку програм Java.

Android Studio є обов’язковим для розробки мобільних програм Android і є офіційним інтегрованим середовищем розробки (IDE) для платформи Android. Android Studio базується на IntelliJ IDEA, написаному на Java IDE, розробленому для максимального підвищення продуктивності завдяки потужному редактору коду та інструментам розробника.

Деякі додаткові функції Android Studio включають швидкий і багатофункціональний емулятор, гнучку систему збирання на основі Gradle, уніфіковане середовище для розробки програм для мобільних пристроїв Android, інструменти Lint і багато іншого.

Gradle — це гнучкий інструмент автоматизації збірки, який працює на віртуальній машині Java (JVM) і потребує комплекту розробки Java. Він використовується Java API у своїй логіці збірки, як-от плагіни та спеціальні типи завдань, і можуть використовувати цей інструмент на різних платформах.

Оскільки Gradle дозволяє використовувати кеш збірки для повторного використання результатів завдань із попередніх запусків, а також інших оптимізацій, він має високу продуктивність.

Так як є частина системи, призначена для роботи з аналітиком та оператором, необхідно створити простий інтерфейс для їх взаємодії. Для цього скористаємось звичним інструментами для створення веб-сторінок – HTML, JS та CSS.

HTML (Hypertext Markup Language) – це набір тегів, які є основою будь-якої сторінки в Інтернеті. З їх допомогою оголошуються заголовки, параграфи, посилання, картинки та інші елементи.

CSS (Cascading Style Sheets) – це невід'ємна і вагома частина процесу розробки сторінок, тому що саме таблиці стилів дозволяють програмно реалізувати розроблений макет інтерфейсу.

JS (JavaScript) – прототипно-орієнтована сценарна мова програмування, що допомагає зробити HTML-розмітку більш інтерактивною [26].

3.2.3 Реалізація отримання даних за допомогою Data Flow. Процес передачі даних було реалізовано за допомогою служби SQL Server Integration Services. SSIS – це інструмент, що дозволяє у зручному вигляді реалізувати інтеграцію, тобто. реалізувати процес перенесення даних з одного джерела до іншого. Цей процес іноді називають ETL (від англ. Extract, Transform, Load – дослівно «вилучення, перетворення, завантаження»).

В інструменті SSIS є служба Data Flow, за допомогою якої було проведено заповнення таблиць вимірів та фактів. Створені потоки даних представлено на рисунку 3.5.

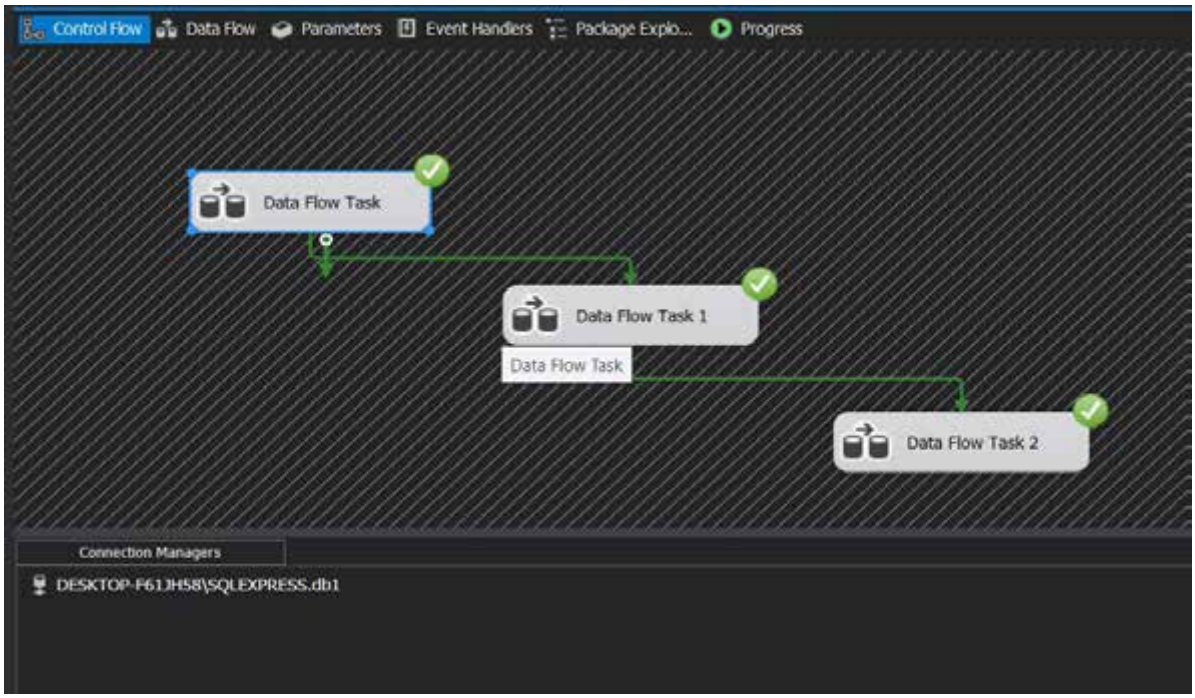

Рис. 3.5 Потоки даних

Розглянемо кожен потік детальніше:

- Перший потік призначений для наповнення даними таблиць вимірів. В якості джерела даних виступає оперативна база даних, з різних сутностей відбираються необхідні дані та заповнюється в таблиці виміри.

- Другий потік призначений створити ключі для таблиці фактів, його структура представлена на рисунку 3.6. На основі заповнених таблиць

вимірив формуються ключові поля таблиці фактів.

- Третій потік призначений для формування числових фактів в таблиці фактів. В якості джерела даних виступає оперативна база даних та таблиці виміри.

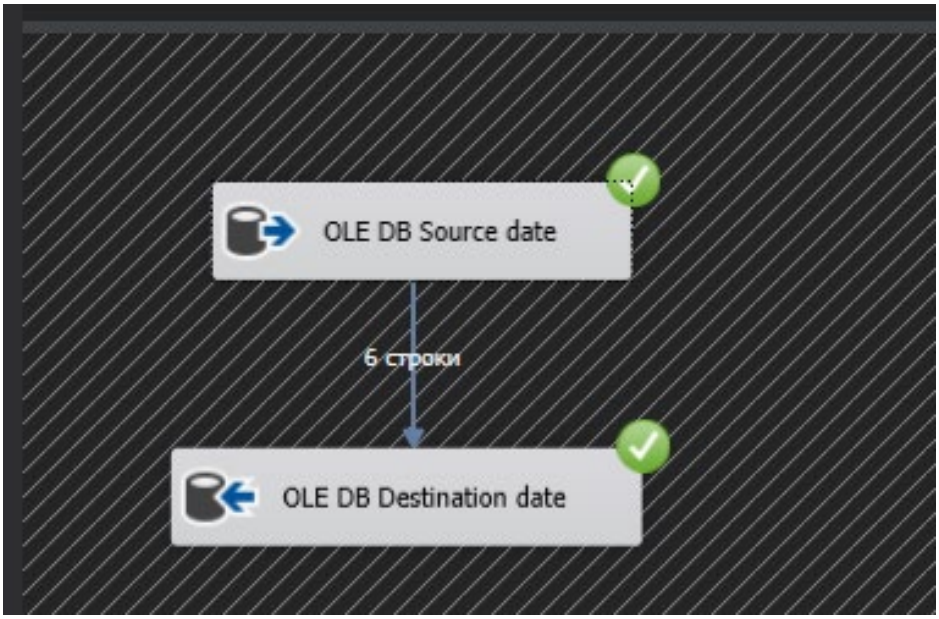

Рис. 3.6 Потік даних 2-го рівня

На рисунку 3.7 представлено процес передачі даних для таблиці DateDim.

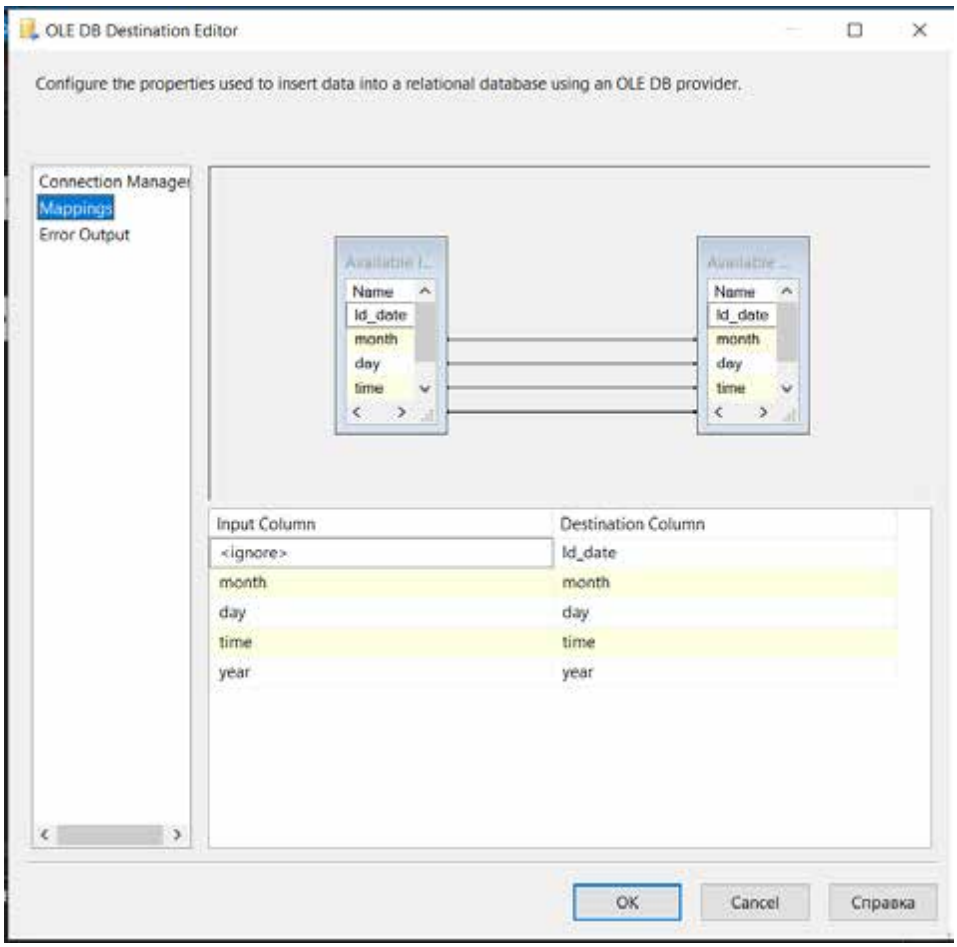

Рис. 3.7 Процес передачі даних

3.2.4 Технології OLAP та Data mining. Data mining (DM) і OLAP є двома поширеними технологіями бізнес-аналітики. BI — це комп'ютерна методологія для ідентифікації та вилучення важливої інформації з бізнес-даних. Інтелектуальний аналіз даних відноситься до галузі інформатики, яка займається вилученням даних, тенденцій і шаблонів із величезних наборів даних. З іншого боку, OLAP розшифровується як Online Analytical Processing — це технологія негайного доступу до даних за допомогою багатовимірних структур. Перш ніж почати, розглянемо різницю між Data mining та OLAP [27].

Data mining означає видобуток знань із величезної кількості даних. Іншими словами, Data mining представляє собою збір інформації та компонування даних із різних областей, таких як сховища даних і алгоритми інтелектуального аналізу даних, пошук тенденцій, шаблонів, які бізнес-організації можуть використовувати для підвищення рівня обслуговування клієнтів, тим самим збільшуючи свій прибуток.

Особливість DM полягає в такий апектах:

- автоматичний пошук шаблонів;
- зосередження на величезних наборах даних і базах даних;
- прогнозування результати;
- створення дієвої інформації.

Система DM може мати схожі основні компоненти, а саме:

- База знань. База знань відноситься до знань про предметну область, які використовуються для оцінки результуючих тенденцій і закономірностей.
- Механізм інтелектуального аналізу даних. Механізм інтелектуального аналізу даних є основним компонентом системи інтелектуального аналізу даних і складається з набору функціональних модулів для різних завдань, наприклад, класифікації, прогнозування, аналізу викидів тощо.
- Модуль оцінки зразків. Модуль оцінки шаблону в основному відповідає за дослідження шаблону за допомогою порогового значення.
- UI (інтерфейс користувача). За допомогою цього модуля користувачі та система аналізу даних спілкуються один з одним.

OLAP – це обчислювальний метод, який дозволяє користувачам отримувати корисну інформацію та запитувати дані, щоб аналізувати їх з різних точок зору. Наприклад, запити бізнес-аналітики OLAP зазвичай допомагають у фінансовій звітності, складанні бюджету, прогнозуванні майбутніх продажів, аналізі тенденцій та інших цілях. Це дозволяє користувачеві аналізувати інформацію бази даних з різних систем баз даних одночасно. Дані OLAP зберігаються в багатовимірних базах даних.

OLAP і DM виглядають схожими, оскільки вони оперують даними, щоб отримати знання, але головна відмінність полягає в тому, як вони працюють з даними. Інструменти OLAP забезпечують багатовимірний аналіз даних і зведення даних.

Основні можливості OLAP:

- підтримка складних обчислень;
- часовий аналіз;
- багатовимірний перегляд даних;
- бізнес-орієнтовані розрахунки;
- гнучка звітність із самообслуговуванням.

3.2.5 Розгортання OLAP-куба. Ядром будь-якої OLAP-системи є OLAP-куба (багатовимірний куб, або гіперкуб). OLAP-структура, створена з робочих даних, називається OLAP-кубом. Він складається з чисельних фактів, розподілених за вимірами. Зазвичай куб створюється за допомогою з'єднання таблиць із застосуванням схеми «зірка», або схеми «сніжинка».

Створення OLAP-куба починається з визначення джерела даних, що стануть основою. Це може бути навіть кілька баз даних, з яких будуть вибиратись лиш деякі необхідні таблиці. В даному випадку, для роботи з OLAP-кубом було розроблено сховище даних. Його підключення представлено на рисунку 3.8. При створені джерела даних важливо вказати, що тип облікових даних – наслідування.

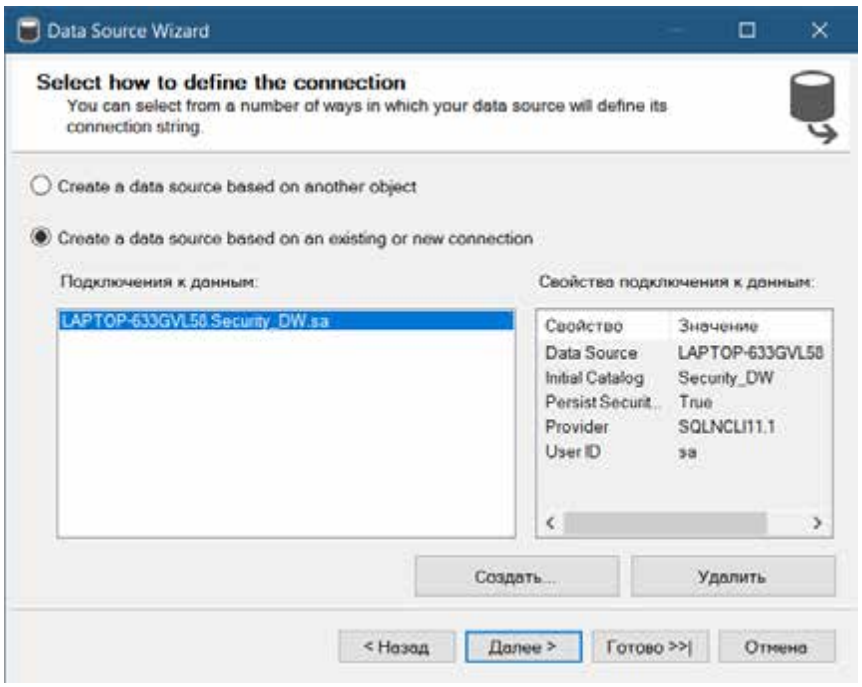

Рис. 3.8 Підключення сховища SecurityDW

На основі створеного джерела даних формується уявленні джерела, що призначене для подальшої роботи з вимірами та самим кубом. Досить важливим аспектом розгортання куба є вірно створені виміри. Саме вони надаю ту багатомірність структури. На рисунку 3.9 представлені сформовані виміри IndicatorDim та AddressDim.

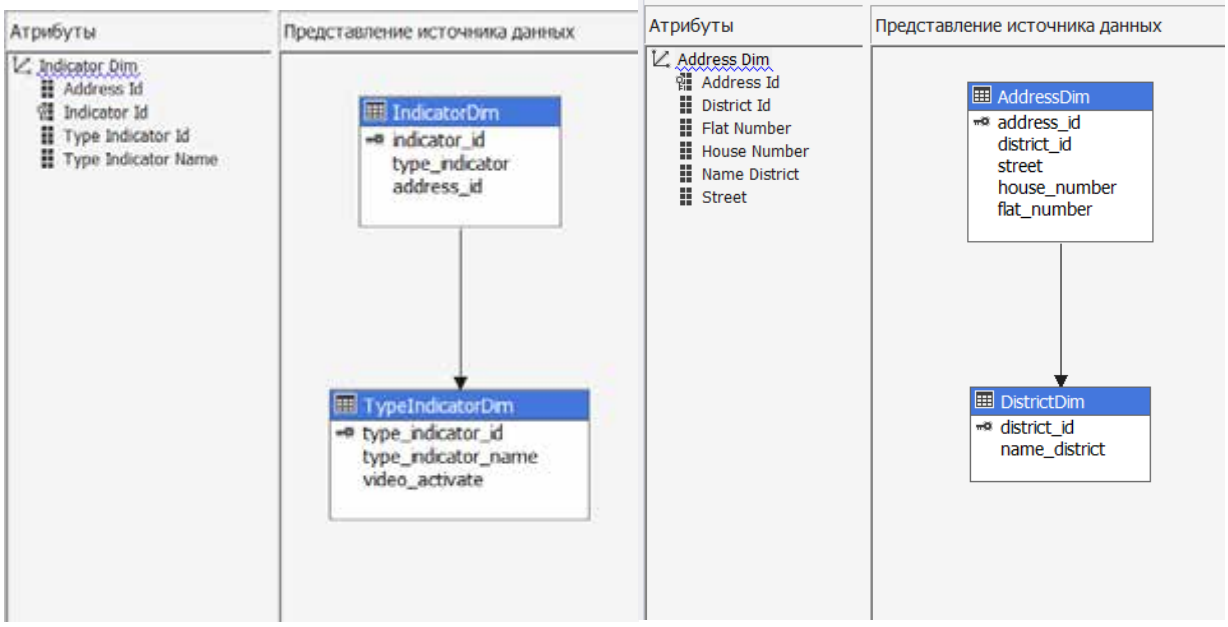

Рис. 3.9 Виміри IndicatorDim та AddressDim

Наступним етапом роботи над OLAP кубом – створення самого куба. Спочатку визначається таблиця фактів, потім визначаються виміри і вже після цього можна розгортати куб. На рисунку 3.10 представлено конструктур з зібраними компонентами куба.

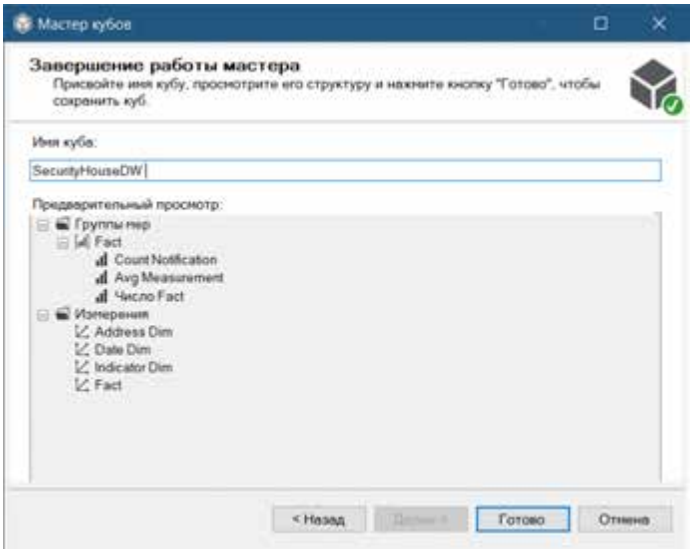

Рис. 3.10 Структура куба

Розгорнутий OLAP куб представлений на рисунку 3.11 має таку ж структуру як і сховище даних. Причина полягає в тому, що в якості джерела даних виступає лише сховище даних, адже на даний момент система має саме таку логіку. В подальшому, структура куба може змінюватись згідно вимог, що надає аналітичному модулю властивість масштабованості та гнучкості.

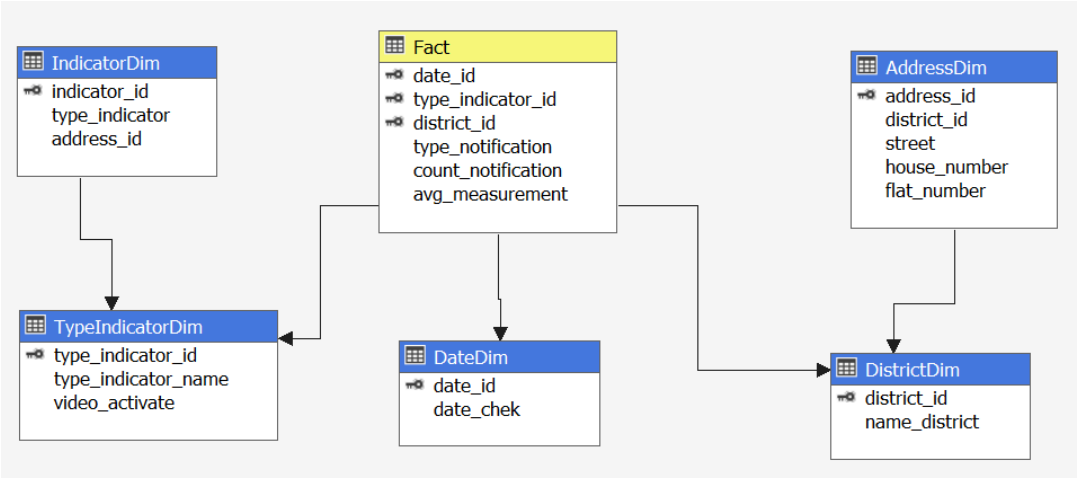

Рис. 3.11 Куб SecurityHouse

3.2.6 Розробка аналітичного модуля системи.

3.2.5.1 Decision Trees Algorithm. Алгоритм Microsoft Decision Trees— це гібридний алгоритм, який включає різні методи для створення дерева та підтримує кілька аналітичних завдань, зокрема регресію, класифікацію та асоціацію. Алгоритм Microsoft Decision Trees має єдиний батьківський вузол, який представляє модель та її метадані. Під батьківським вузлом знаходяться незалежні дерева, які представляють передбачувані атрибути, що були обрані [28].

Особливість застосування даного алгоритму полягає в універсальності. При єдиному набору даних існує можливість обрати різні поля прогнозування і алгоритм створить інше дерево рішень. Дерево для кожного передбачуваного атрибута містить інформацію, яка описує, як вибрані стовпці введення впливають на результат цього конкретного передбачуваного атрибута.

Алгоритм Microsoft Decision Trees також може містити лінійні регресії у всьому дереві або його частині. Якщо атрибут, який прогнозується, є безперервним числовим типом даних, модель може створити вузол дерева регресії, де зв’язок між атрибутами можна моделювати лінійно. У цьому випадку вузол містить формулу регресії.

На основі розробленого раніше SecurityHouse куба розгортається нова структура на основі алгоритму Microsoft Decision Trees. Цей процес представлено на рисунку 3.12.

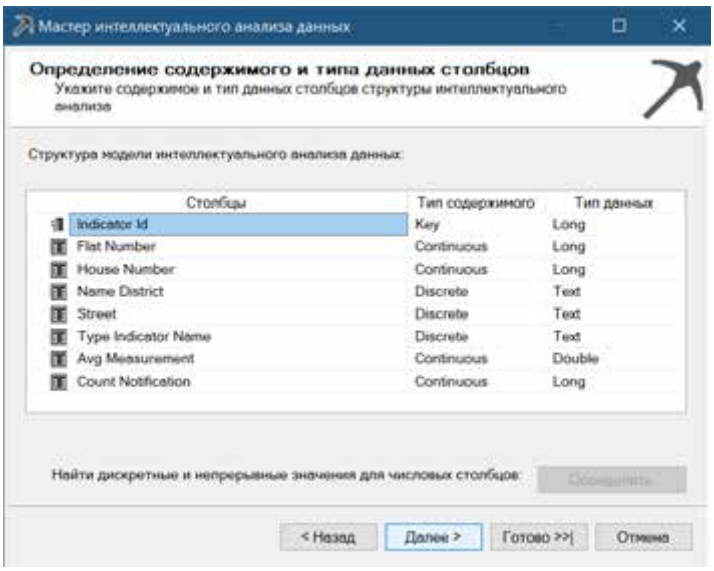

Рис. 3.12 Структура DTree

3.2.5.2 Naive Bayes Algorithm. Алгоритм Microsoft Naive Bayes — це набір алгоритмів класифікації на основі теореми Байєса . Наївна модель Байєса має єдиний батьківський вузол, який представляє модель та її метадані, а під цим батьківським вузлом — будь-яку кількість незалежних дерев, які представляють передбачувані атрибути, які були обрані [29].

Для кожного передбачуваного атрибута та значення модель виводить дерево, яке містить інформацію, що описує, як різні вхідні стовпці вплинули на результат цього конкретного передбачуваного. Кожне дерево містить передбачуваний атрибут і його значення, а потім серію вузлів, які представляють вхідні атрибути

Оскільки наївна модель Байєса не дозволяє безперервні типи даних, усі значення вхідних стовпців розглядаються як дискретні або дискретизовані. Структура даних для алгоритму представлена на рисунку 3.13.

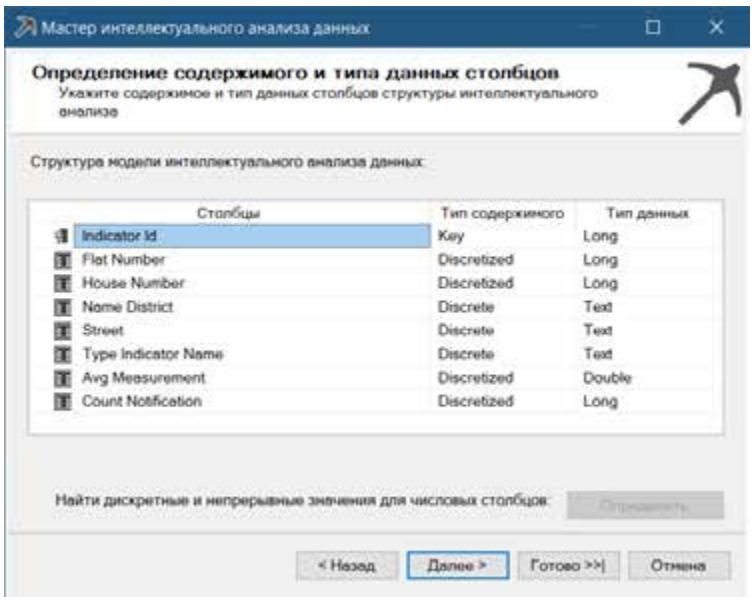

Рис. 3.13 Структура NBayes

3.2.5.4 Clustering Algorithm. Алгоритм кластеризації Microsoft — це алгоритм сегментації або кластеризації, який перебирає випадки в наборі даних, щоб групувати їх у кластери, які містять подібні характеристики. Модель кластеризації має просту структуру. Кожна модель має один батьківський вузол, який представляє модель та її метадані, і кожен батьківський вузол має плоский список кластерів [30].

Кожен дочірній вузол представляє один кластер і містить детальну статистику про атрибути випадків у цьому кластері. Це включає підрахунок кількості випадків у кластері та розподіл значень, які відрізняють кластер від інших кластерів. Батьківський вузол містить корисну статистику, яка описує фактичний розподіл усіх випадків навчання.

На основі даного алгоритма була розгорнута структура Clust, що має таку ж структуру як і DTree.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Інтерфейс мобільного додатку

Система моніторингу периметру безпеки розумного будинка з мобільного пристрою складається з кількох різних інтерфейсів, призначених для роботи з різними типами користувачів. Розглянемо макет інтерфейсу мобільного застосунку, що має взаємодіяти з основною частиною цільової аудиторії розроблюваної системи.

Складність розробки інтерфейсу полягає в ефективному ти зручному представлені всього доступного функціонала на невеликій екранній площині мобільного телефону. На рисунку 4.1 представлено макет головного меню застосунку.

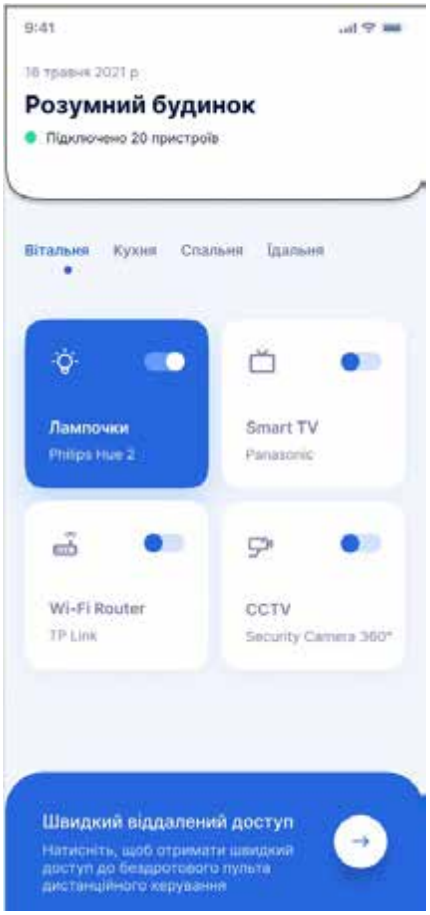

Рис. 4.1 Головне меню мобільного застосунку

Представлене меню має кілька зон, основна – робоча, знаходить по центрі та має зручне розташування всіх елементів в досяжності пари кліків, майже без зміни положення руки. Центральна частина відразу поділена на кімнати будинку та демонструє підключені пристрої до системи, надаючи можливість керувати ними в один дотик.

На рисунку 4.2 наведено макет сторінки статистики та маніпуляцій підключеного пристрою до мережі.

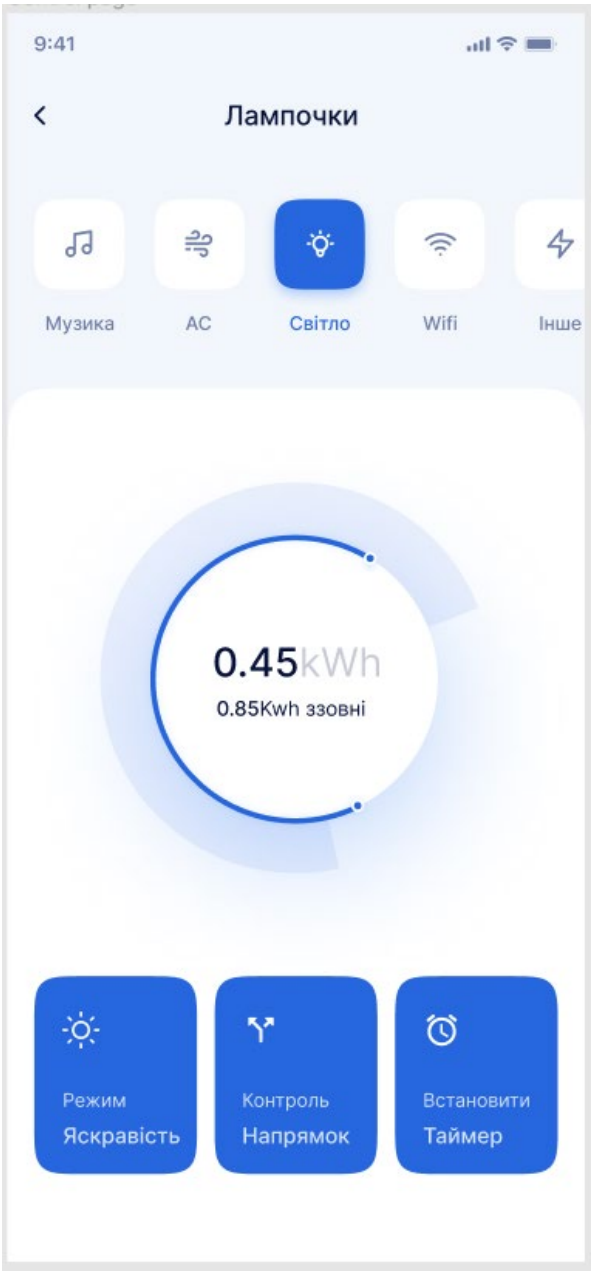

Рис. 4.2 Споживання світла конкретним пристроєм

На представленій вище сторінці вже кілька зон взаємодії, адже центральну позицію займає саме інформаційний блок, що може демонструвати скільки та котрого ресурсу було спожито пристроєм для забезпечення бажаної функціональності. Також, обираючи з нижнього меню маніпуляцій, на центральну зону виводяться необхідні індикатори, що надають можливість більш делікатніше вносити коригування роботи пристрою. У верхній частині представлено список різних девайсів, що підключені до мережі та надають можливість керувати ними віддалено.

На рисунку 4.3 представлено одна з найбільш корисних сторінок мобільного застосунку – звітні спожитого енергоресурсу, отриманого від лічильника.

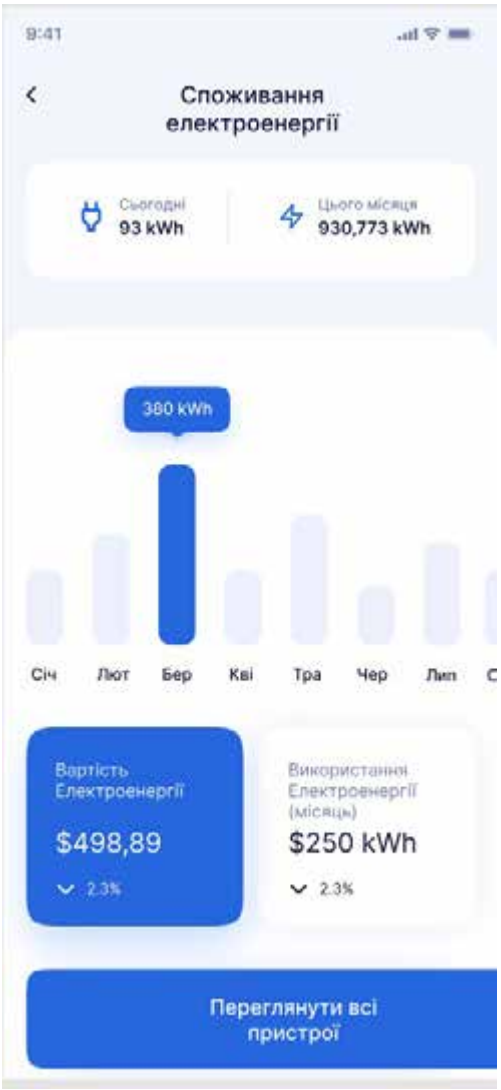

Рис. 4.3 Сторінка споживання електроенергії

Представлена сторінка має більш статистичний характер, що дозволяє переглядати спожиті ресурси в розрізі місяців.

4.2 Результати аналітичного модуля

4.2.1 Decision Trees Algorithm. Розгортанням структури DTree – візуалізовано дерево рішень, що представлено на рисунку 4.4.

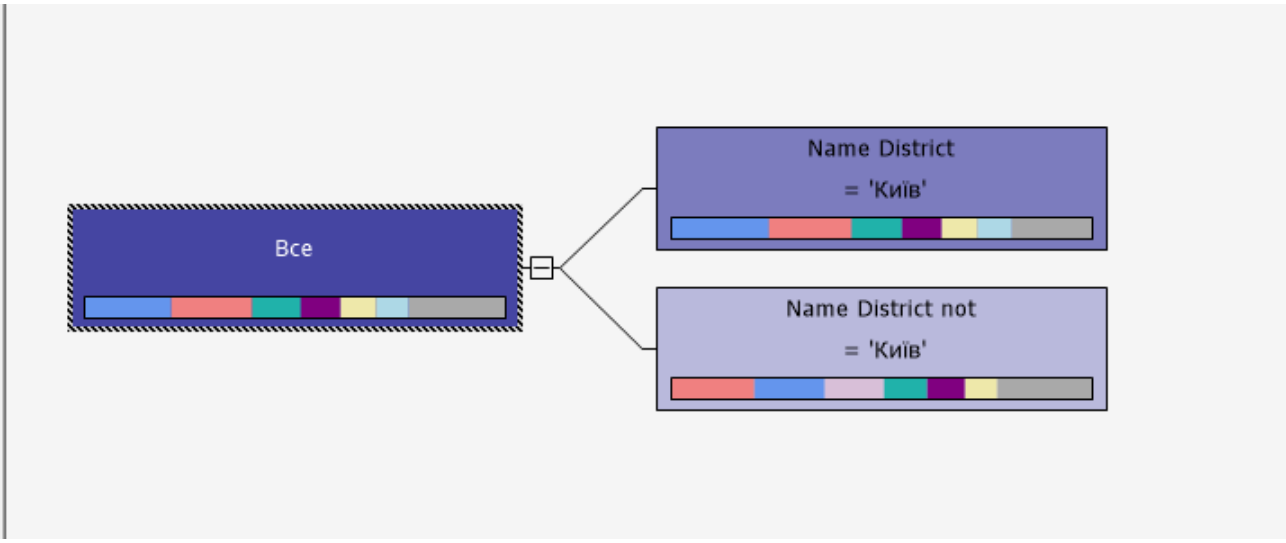

Рис. 4.4 Дерево рішення для прогнозування Type Indicator Name

Загалом структура інтелектуального аналізу знайшла залежність від району будинку. Отже DTree має розбиття на два вузли. Загальний вузол демонструє загальний розподіл даних про тип датчиків, що представлено на рисунку 4.5.

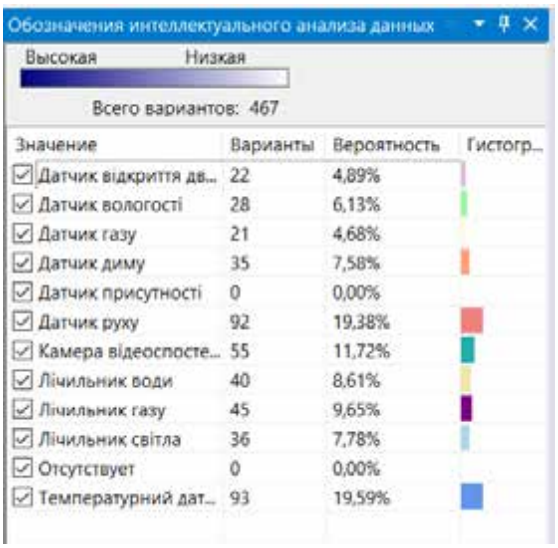

Рис. 4.5 Загальний розподіл значень Type Indicator Name

З наведеного розподілу можна зробити висновки, що найбільшою популярністю користуються сенсори руху та термометри.

Розглянемо наступний вузол, що містить значення з району під назвою Київ. На рисунку 4.6 представлено розподіл значень по типу датчика в київських будинках.

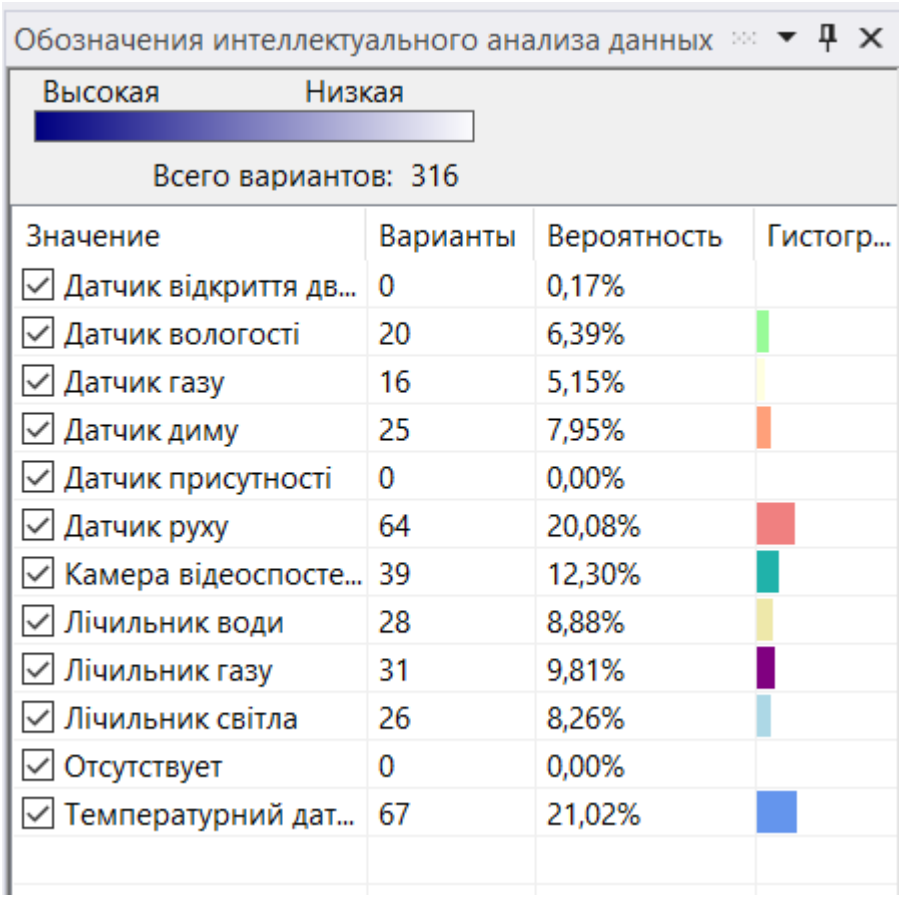

Рис. 4.6 Розподіл значень Type Indicator Name в Києві

З представленого розподілу видно, що, за умови надходження даних з будинків, розташованих в Києві, то найімовірніше це буде звітне повідомлення від датчика руху(20%) або температурного покажчика(21%). Вірогідність надходження повідомлень щодо лічильник, раз на місяць, складає близько 8-10% для кожного.

Розглянемо інший вузол, що відповідає всім іншим випадкам надходження даних не з Києва, що представлено на рисунку 4.7.

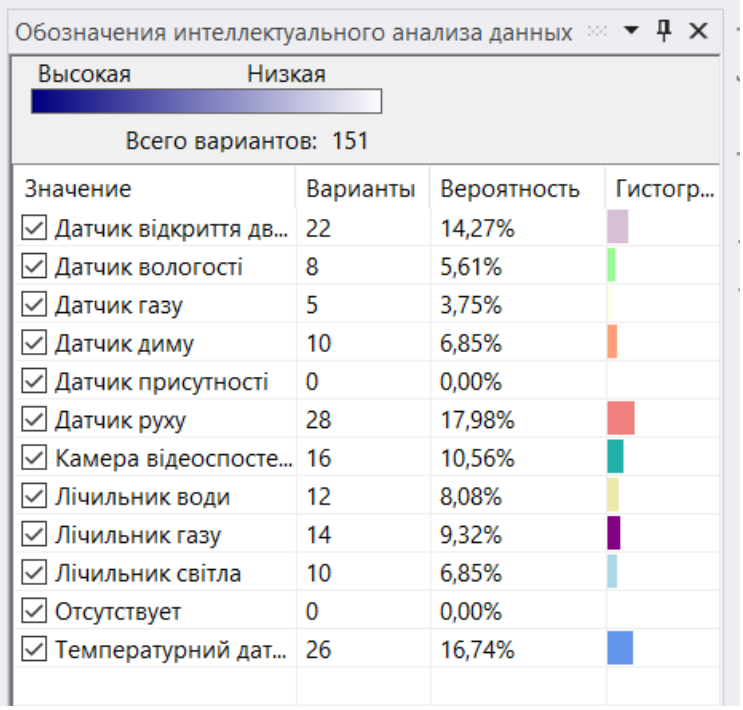

Рис. 4.7 Розподіл значень Type Indicator Name не в Києві

З наведених результатів можна сказати у випадку, коли повідомлення надходить не з Києва, то найімовірніше це буде від датчика температури, руху чи відкриття вікон/дверей, адже вони мають близькі значення вірогідності – 15%.

4.2.2 Naive Bayes Algorithm. Структура інтелектуального аналізу NBayes представлена як мережа залежностей, рисунку 4.8.

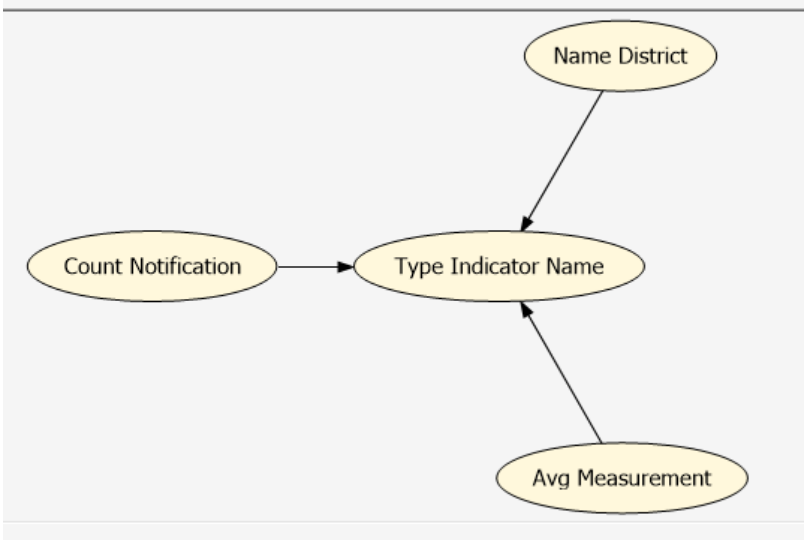

Рис. 4.8 Мережа залежностей NBayes

Вище представлено мережу залежностей, в центрі якої знаходиться прогнозоване значення та його взаємозв'язки з іншими даними. Структура виявила залежності між значеннями типу індикатора, району його застосування та середнім значенням їх показників.

4.2.4 Clustering Algorithm. Після обробки структури інтелектуального аналізу Clust отримано модель даних, що представлено на рисунку 4.9.

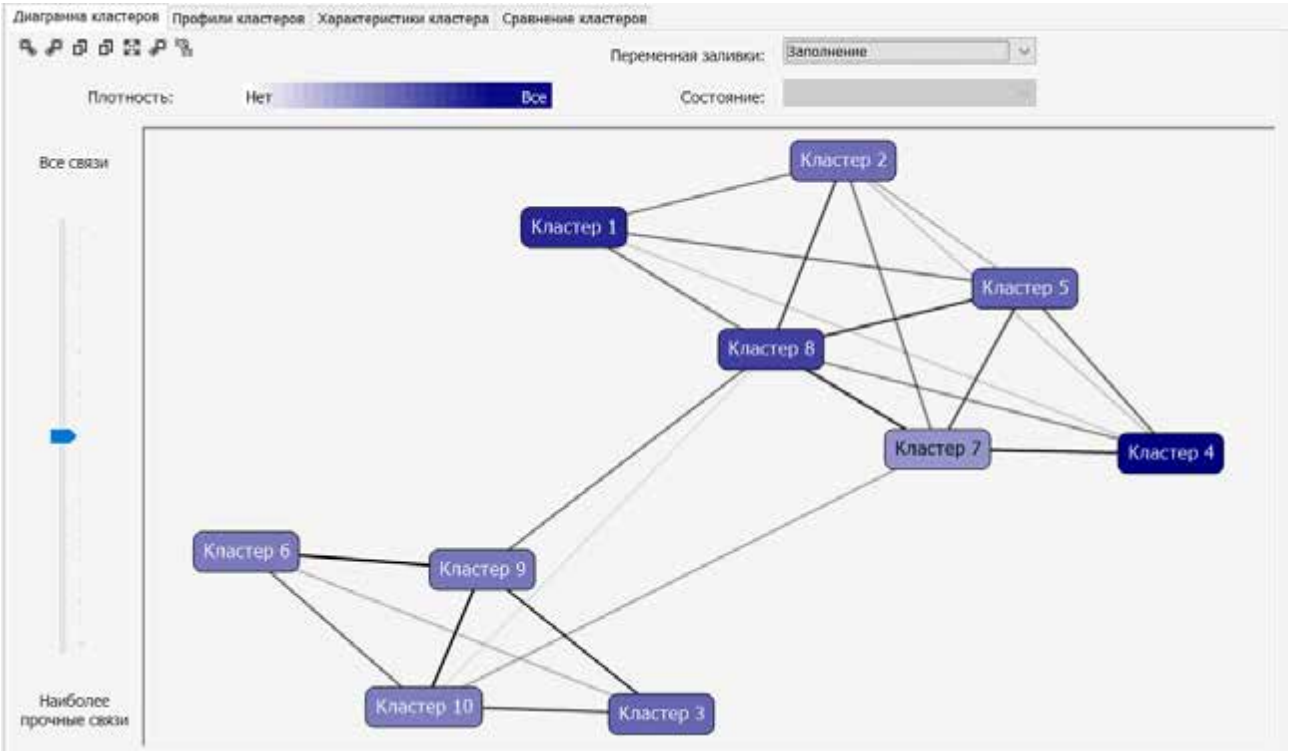

Рис. 4.9 Структура моделі Clust

З наведеного зображення, можна зрозуміти, що найбільша кількість інформації зосереджена саме в кластерах 1, 8 та 4. Розглянемо їх більш детально:

- Кластер 1. Дані належать будинкам розташованим в Києві та надійшли від таких датчиків:
  - температурний датчик – 41,08%;
  - датчик руху – 41,08%;
  - датчик диму – 4,57%;
  - камера відеоспостереження – 4,36%.
- Кластер 8. Дані належать будинкам розташованим в Києві та надійшли

від таких датчиків:

- лічильник газу – 21,39%;
- лічильник світла – 18,89%;
- камера відеоспостереження – 14,92%;
- датчик руху – 14,27%;
- датчик диму – 12,87%;
- лічильник води – 12,68%.

- Кластер 4. Дані належать будинкам розташованим в Києві та надійшли від таких датчиків:

  - лічильник газу – 31,55%;
  - лічильник світла – 22,07%;
  - датчик диму – 13,40%;
  - лічильник води – 13,13%.

Загалом кластери, виявили, що популярністю датчики відкриття вікон/дверей користуються в приватних будинках, датчики вологості ж, навпаки, більше встановлюють в квартирах.

ВИСНОВКИ

В ході роботи були використані такі технології та інструменти: Realm, MS SQL Server, HTML5, CSS, JS, Java, OLAP та Data Mining. Наведено детальний аналіз моніторингу периметра безпеки розумного будинку з мобільного пристрою.

В результаті було створено структурно-функціональні та об'єктно-орієнтовані моделі, що в повному обсязі розкривають предметну область моніторингу периметра безпеки розумного будинку з мобільного пристрою.

Наступним етапом при розробці надбудови стало створення інформаційного забезпечення системи, складовими якого є розроблені логічна модель даних, фізична реалізація бази даних та сховище даних. Унікальність даного інформаційного забезпечення полягає саме в застосуванні технологій пов'язаних зі сховищем даних, що являє собою основою для подальшої розробки аналітичного модуля.

Завершальним етапом розробки програмного забезпечення стало створення модулів для роботи з виконавчим провадженням та аналітичного модуля. Архітектура системи представляє собою клієнт-серверну топологію. Аналітичний модуль створюється на основі OLAP-технологій за допомогою інструментів Data mining, що використовують сховище даних. В останньому розділі представлено результати дослідження.

В ході розгортання даних інтелектуальних структур та аналізу отриманих результатів було виявлено такі особливості даних:

- Decision Tree: в результаті розгортання структури з прогнозуванням типу індикатора, що відправить повідомлення, було отримано такі дані: розбиття інформації на два вузли за назвою районів в яких розміщені індикатори. Отже якщо повідомлення надійшло від Києва то найімовірніше це буде звітне повідомлення від датчика руху(20%) або температурного покажчика(21%). У випадку, коли повідомлення надходить не з Києва, то найімовірніше це буде від датчика

температури, руху чи відкриття вікон/дверей, адже вони мають близькі значення вірогідності – 15%.

- Naive Bayes: структура виявила залежності між значеннями типу індикатора, району його застосування та середнім значенням їх показників.

- Cluster: структура представила кластери, що загалом характеризують дані системи, а саме: популярністю датчики відкриття вікон/дверей користуються в приватних будинках, датчики вологості ж, навпаки, більше встановлюють в квартирах.

Загалом в ході дослідження були виявлені, як і природні взаємозв’язки між даними, наприклад споживання ресурсів згідно часового періоду, так менш помітні залежність типу індикатора від району розміщення будинку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is a Smart Home? Everything You Need to Know|Definition from TechTarget. [Електронний ресурс] URL: https://www.techtarget.com/iotagenda/definition/smart-home-or-building

2. Розумний будинок: комфорт та ефективність в єдиній системі. [Електронний ресурс] URL: https://www.vodafone.ua/shop/ua/blog/rozumnij-budinok-komfort-ta-efektivnist-v-edinij-sistemi.html

3. Smart Home-Control and Monitoring System Using Smart Phone. [Електронний ресурс] URL: https://www.researchgate.net/publication/250306737_Smart_Home-Control_and_Monitoring_System_Using_Smart_Phone

4. Monitoreal Video Security Assistant Device - Monitoreal. [Електронний ресурс] URL: https://monitoreal.com/security-assistant-device/

5. Introducing Monitoreal Secure Guard AppYour Comprehensive Surveillance Solution on the Go!. [Електронний ресурс] URL: https://monitoreal.com/homeowners/introducing-monitoreal-secure-guard-app-your-comprehensive-surveillance-solution-on-the-go/

6. Smart Home 101. [Електронний ресурс] URL: https://www.iotforall.com/smart-home-101

7. See smart home devices that work with Google | Google Home. [Електронний ресурс] URL: https://home.google.com/what-is-google-home/

8. See smart home devices that work with Google | Google Home. URL: https://home.google.com/what-is-google-home/

9. Explore what you can do with Google Nest or Home devices. [Електронний ресурс] URL: https://support.google.com/googlenest/answer/7130274?hl=en

10. Hubitat Unveils Mobile App for Hubitat Elevation Home Automation Platform. [Електронний ресурс] URL: https://www.prnewswire.com/news-

releases/hubitat-unveils-mobile-app-for-hubitat-elevation-home-automation-platform-300856291.html

11. Hubitat Elevation Mobile App. [Електронний ресурс] URL: https://docs2.hubitat.com/mobile-app

12. Structured Analysis and Design Technique and Data Flow Diagrams. [Електронний ресурс] URL: http://r.web.umkc.edu/rer554/cs457_homework5.html

13. What is a Data Flow Diagram. [Електронний ресурс] URL: https://www.lucidchart.com/pages/data-flow-diagram

14. Introduction to Object-Orientation and the UML. [Електронний ресурс] URL: https://agiledata.org/essays/objectorientation101.html

15. What is Unified Modeling Language (UML)?. [Електронний ресурс] URL: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/

16. UML Use Case Diagram Tutorial. [Електронний ресурс] URL: https://www.lucidchart.com/pages/uml-use-case-diagram

17. What is Activity Diagram? [Електронний ресурс] URL: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/

18. UML Activity Diagram Tutorial [Електронний ресурс] URL: https://www.lucidchart.com/pages/uml-activity-diagram

19. UML Sequence Diagram Tutorial. [Електронний ресурс] URL: https://www.lucidchart.com/pages/uml-sequence-diagram

20. Structured Analysis and Design Technique and Data Flow Diagrams [Електронний ресурс] URL: http://r.web.umkc.edu/rer554/cs457_homework5.html

21. An Introduction to Realm Database [Електронний ресурс] URL: https://medium.com/excellentweb/an-introduction-to-realm-database-2881f0f8c231

22. Сховище даних, OLAP - куб [Електронний ресурс] URL: http://surl.li/muwoj

23. Ситник Н. В. Організація баз та сховищ даних [Електронний ресурс] : практикум / Н.В. Ситник. — К. : 2017. — 148 с. ISBN 978–966–926–212–7

24. Клієнт-серверна архітектура та ролі серверів. [Електронний ресурс] URL: http://surl.li/jbkid

25. Java Mobile Applications Development: What You Need to Know. [Електронний ресурс] URL: https://www.webiotic.com/java-mobile-applications-development-what-you-need-to-know/

26. Что такое HTML, CSS, PHP и JS? [Електронний ресурс] URL: https://wayup.in/ua/blog/chto-takoe-html-css-php-i-js

27. Difference between Data Mining and OLAP [Електронний ресурс] URL: https://www.javatpoint.com/data-mining-vs-olap

28. Mining Model Content for Decision Tree Models (Analysis Services - Data Mining) [Електронний ресурс] URL: http://surl.li/mumlr

29. Mining Model Content for Naive Bayes Models (Analysis Services - Data Mining) [Електронний ресурс] URL: http://surl.li/mumpi

30. Mining Model Content for Clustering Models (Analysis Services - Data Mining) [Електронний ресурс] URL: http://surl.li/mumpr