

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

УДК 004.9:17.021.1
«ПОГОДЖЕНО» «ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»
Декан факультету інформаційних технологій Завідувач кафедри комп'ютерних наук

НУБІП України

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

2021р. 30 листопад 2021р.

НУБІП України

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Дорадча система виявлення екстремальних ситуацій на основі моніторингу поведінки людини

Спеціальність 122 Комп'ютерні науки

Освітня програма Інформаційні управляючі системи та технології

Орієнтація освітньої програми _____ освітньо-професійна _____

Гарант освітньої програми
доктор техн.н. професор (науковий ступінь та вчене звання) Бондаренко В.Є. (ПІБ)

НУБІП України

Керівник магістерської кваліфікаційної роботи

Старший викладач

Бородкін Г.О.

Виконала _____ (ПІБ студента)
Шрамко В.І. (ПІБ студента)

НУБІП України

КИЇВ-2021

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

НУБІП України

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук
 к.т.н., доцент Голуб Б.І.
 (науковий ступінь, вчене звання) (підпис) (ПІБ)
 «29» жовтня 2020 року

НУБІП України

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ
Шрамко Вікторії Ігорівни
 (прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»

Освітня програма Інформаційні управляючі системи і технології

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Дорадча система виявлення екстремальних ситуацій на основі моніторингу поведінки людини

затверджена наказом ректора НУБІП України від «29» жовтня 2020 р. №1634 «С»

Термін подання завершеної роботи на кафедру 30 листопада 2021 р.

Вихідні дані до магістерської кваліфікаційної роботи

Перелік питань, що підлягають дослідженню:

№ з/п	Питання, що підлягає дослідженню	Строк виконання	Примітка
1.	Системний аналіз предметної області.	01.09.2021	
2.	Моделювання системи	10.09.2021	
3.	Розробка системи	17.09.2021	
4.	Результати дослідження	29.10.2021	
5.	Попередній захист	30.11.2021	
6.	Захист	14.12.2021	

Дата видачі завдання «29» жовтня 2020 р.

Керівник магістерської кваліфікаційної роботи Бородкін Г.О.
 (підпис) (прізвище та ініціали)

Завдання прийняв до виконання Шрамко В.І.
 (підпис) (прізвище та ініціали студента)

НУБІП України

НУБІП України

НУБІП України

ЗМІСТ

НУБІП України

ВСТУП

4

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

6

2. МОДЕЛЮВАННЯ СИСТЕМИ

11

НУБІП України

3 РОЗРОБКА СИСТЕМИ

29

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

45

ВИСНОВОК

48

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

49

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВСТУП

Аналіз аномальної поведінки людини служить сигналом попередження перед екстремальними ситуаціями. Однак більшість виявлення аномальної поведінки людини наразі покладається на ручний моніторинг. Цей метод критикують за суб'єктивність і відсутність своєчасності. Тому пропонуємо розглянути багатетапний метод аналізу аномальної поведінки людини за допомогою автоматизованої дорадчої системи.

Аналіз поведінки, вперше запроваджений психологією, використовує психологічні знання для опису, аналізу, прогнозування та контролю зовнішніх проявів особистості.

При виявленні поведінки людини групову поведінку людей важко описати. Немає точного визначення аномальної поведінки людини. Про так звану аномальну поведінку судять за відхиленнями від характеристик групової поведінки.

Об'єкт дослідження: процес виявлення екстремальних ситуацій на основі моніторингу поведінки людей

Предмет дослідження: дорадча система виявлення екстремальних ситуацій на основі моніторингу поведінки людей.

Мета: перевірка доцільності використання технологій OLAP та Data Mining для виявлення екстремальних ситуацій на основі моніторингу поведінки людей шляхом розробки дорадчої системи. Для досягнення мети дослідження було поставлено та вирішено такі основні задачі:

1. проаналізувати основні поняття за напрямом дослідження;
2. розкрити завдання та функції дорадчої системи виявлення екстремальних ситуацій;
3. дослідити методи моніторингу поведінки людей;
4. проаналізувати технології OLAP та Data Mining;
5. визначити шляхи удосконалення автоматизованих систем для моніторингу поведінки людей.

НУБІП України

Практичний результат роботи: проведено дослідження дорадчої системи виявлення екстремальних ситуацій, використано технології OLAP

іта Data Mining для аналізу та вивчення системи. Реалізоване рішення

дозволяю отримувати звіти інформативні звіти для аналітики.

НУБІП України

Проект складається з 4 розділів.

Перший розділ містить огляд про системний аналіз предметної області.

Опис системи раннього попередження, математична модель екстремальних ситуацій.

НУБІП України

У другому розділі здійснюється опис етапів моделювання системи за допомогою об'єктно-орієнтованого підходу.

Третій розділ має описання архітектури системи, що досліджується та усі складові, які представлені в ній.

НУБІП України

У четвертому розділі представленні результати дослідження магістерської роботи. Сформовані звіти їх аналіз та опис.

НУБІП України

НУБІП України

НУБІП України

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.

НУБІП України

Системи раннього попередження є ключовими елементами зменшення

ризиків катастроф і спрямовані на уникнення або зменшення збитків,

спричинених небезпеками. Щоб бути ефективними, системи раннього

попередження повинні активно залучати людей і громади, які піддаються

ризиків від низки небезпек, сприяти освіті та поінформованості населення про

ризиків, ефективно поширювати повідомлення та попередження та

забезпечувати постійний стан готовності та ранні дії. Значення ефективної

системи раннього попередження полягає у визнанні її переваг місцевим

населенням. Системи раннього попередження про ризиків повинні спиратися

на надійну науково-технічну основу та орієнтуватися на людей або сектори,

які найбільше піддаються ризику. Це передбачає прийняття системного

підходу, що включає всі відповідні фактори ризику, пов'язані з небезпечною

клімату чи соціальною вразливістю, а також з короткостроковими чи

довгостроковими процесів.

Системи раннього попередження включають виявлення, аналіз,

прогнозування, а потім розповсюдження попереджень з подальшим

прийняттям та впровадженням рішень. Такі системи існують у багатьох

частинах світу для моніторингу, прогнозування та попередження людей про

небезпеки. Щоб бути ефективною та повною, система раннього

попередження має містити чотири взаємодіючих елементи, а саме:

- знання про ризиків;
- моніторинг та попередження;
- поширення та комунікація;
- здатність реагувати.

У Європі є значний досвід використання систем раннього попередження, особливо щодо ризику природних надзвичайних ситуацій.

Системи раннього попередження мають безпосереднє відношення до

НУБІП України

різноманітних секторів, на які в першу чергу впливають ризики, пов'язані з кліматом, як-от здоров'я, зменшення ризику катастроф.

Проблема ефективного і своєчасного реагування на надзвичайні ситуації, а також оптимальний розрахунок сил і засобів, необхідних для їх ліквідації, останнім часом набуває особливої актуальності.

Тяжкість ситуації спричинена через недостатньо ефективне планування надзвичайних ситуацій.

Мета системи раннього попередження, орієнтованої на людей, полягає в тому, щоб дати можливість особам і громадам, яким загрожує небезпека,

мати достатньо часу та належним чином зменшити можливість травмування, втрати життя та пошкодження майна та навколишнього середовища, готовності та здатності реагувати. Системи раннього

попередження найкращої практики також мають міцні взаємозв'язки та

ефективні канали зв'язку між усіма елементами. Знання про ризик, що виникають внаслідок комбінації небезпек та вразливостей у певному місці.

Оцінка ризиків допомагають мотивувати людей, розставляти пріоритети потреб системи раннього попередження та керувати підготовкою до

запобігання катастрофам та реагування на них. Служба моніторингу та

попередження. Сервіси попередження лежать в основі системи. Повинна бути надійна наукова основа для передбачення та прогнозування небезпек та надійного прогнозування та попередження функціонує 24 години в день.

Постійний моніторинг параметрів небезпеки та прекурсорів є важливим для

своєчасного отримання точних попереджень. Служби попередження про різні небезпеки слід координувати, де це можливо, щоб отримати переваги спільних інституційних, процедурних і комунікаційних мереж.

Характерні ознаки екстремальних ситуацій, такі як раптовість

виникнення, швидкість розвитку, неповнота та невизначеність вихідної

інформації, різноманітність і ланцюговий характер наслідків ускладнюють використання традиційних емпіричних методів для їх вивчення.

Попередження щодо поширення та комунікації повинні охоплювати тих, хто знаходиться в групі ризику. Чіткі повідомлення, що містять просту корисну інформацію, є вирішальними для забезпечення належних відповідей, які допоможуть захистити життя та засоби до існування.

У цьому контексті математичне моделювання все частіше використовується для аналізу та прогнозування надзвичайних ситуацій у багатьох випадках є єдино допустимим, наприклад, перевірка небезпечних природних або техногенні явища.

Математична модель екстремальних ситуацій називається система відносин, рівняння, нерівності, відтворити геометричні терміни тощо, які відображаються в математичній формі або імітує найважливіші функції та властивості реальних небезпечних явищ для цієї мети аналіз та прогноз їх виникнення, розвиток та наслідки.

Особливості математичної моделі багато в чому визначаються типом екстремальної ситуації, що моделюється.

Усі надзвичайні ситуації можна поділити на надзвичайні ситуації соціально-політичного, природного та техногенного характеру.

Природня включають землетруси, виверження вулканів, цунамі, повені, урагани, лавини, зсуви, посухи, лісові пожежі та ін.

Технологічні надзвичайні ситуації пов'язані з аваріями на електростанціях та промислових підприємствах, а також з аваріями на транспорті, пов'язаними з вибухами, пожежами, хімічним та радіоактивним забрудненням території.

Соціально-політичні надзвичайні ситуації включають: війни, прикордонні конфлікти, тероризм, саботаж.

Комбіновані природно-технологічні та природно-соціальні надзвичайні ситуації включають спад, епідемії та хвороби тварин.

Інфекційні хвороби тварин, інфекційні хвороби сільськогосподарських культур.

Всі перелічені вище надзвичайні ситуації можуть бути досліджено методами математичного моделювання.

Створення математичної моделі аварійних ситуацій складається з кількох етапів.

На першому етапі, потрібний змістовний опис екстремальної ситуації на основі всіх наявних знань і результатів польових досліджень подібних ситуацій, консультації з фахівцями, вивчення спеціальної літератури.

На другому етапі формалізація змістовного опису моделі, математична постановка задачі з деталями всіх необхідні вихідні дані та необхідні значення.

На третьому етапі формалізована схема екстремальної ситуації слід перенести на її математичну модель. Для цього має бути включена вся доступна інформації, взаємозв'язки, нерівності, рівняння, алгоритми.

На четвертому етапі досліджується сама модель. За допомогою багатовимірних розрахунків досліджуються властивості моделі та її поведінка у різних умовах.

Наступним кроком буде застосування моделі до опису реальних надзвичайних ситуацій. Шляхом зіставлення результатів обчислювальних експериментів з існуючими експериментальними даними.

Виявлення чи уточнення параметрів моделі, її тестування, налагодження та перевірка адекватності. Необхідною умовою отримання досить точної та надійної математичної моделі аварійних ситуацій є перевірка її доцільності.

Вирішення завдань автоматизації процесів управління в надзвичайних ситуаціях відбувається без комплексного підходу врахування факторів, що передують виникненню екстремальних ситуацій і впливають на їх подальший розвиток. Є такий комплексний підхід впровадження в розробку

дорадчої системи підтримки прийняття рішень щодо попередження надзвичайних ситуацій. У цьому випадку є дві основні варіанти впровадження

дорадчої системи, впровадження єдиної інтегрованої системи, впровадження локальних систем.

Дорадча система в свою чергу - це лише частина автоматизованої системи управління інформацією про попередження, що є в ширшому розумінні сукупністю автоматизованих систем і технічних засобів,

об'єднаних в один комплекс. Програмне забезпечення відіграє ключову роль у його роботі. Цьому сприяють такі фактори, як тенденція до підвищення ступеня універсальності програмного забезпечення та незалежності від

прикладних апаратних технологій, а також створення систем на основі

мікросервісів, коли велика інформація клієнт-сервер система являє собою набір найпростіших модулів, які взаємодіють один з одним обробляти запити користувача самостійно.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

2 МОДЕЛЮВАННЯ СИСТЕМИ

НУБІП України

Моделювання являє собою основний метод дослідження в усіх

областях знань та науково обґрунтованим методом оцінки характеристик складних систем, що використовується для прийняття рішень. Модель – це об'єкт або опис об'єкту, оригінальної системи, що буде замінена другою системою при визначених умовах, гіпотезах для вивчення або відтворення її

окремих властивостей. Під моделюванням треба розуміти процес побудови,

вивчення та застосування моделі. Процес моделювання повинен включати побудову абстракцій, гіпотез та їх конструювання. Модель є інструментом пізнання, який дослідник ставить між собою та об'єктом і завдяки чому

вивчає цікавий йому об'єкт. Існуючі системи та системи, що тільки

проектуються можна ефективно дослідити за допомогою математичних моделей (аналітичних та імітаційних), що в свою чергу реалізуються на сучасних ЕОМ, які в цьому випадку виступають в якості інструмента експериментатору з моделлю системи. Ціль моделювання в тому, щоб

стандартизувати етапи роботи над проектом.

При цьому в процесі моделювання системи важливо виявити ясну і відносно просту внутрішню структуру. Моделювання повинне враховувати облік суперечливих вимог.

Моделювання починається з формування предмету дослідження – системи понять, що відображає суттєві для моделювання характеристики об'єкту.

Одним з поширених видів моделювання в даний час є моделювання з застосуванням різних інформаційних систем, що побудовані з використанням комп'ютерних моделей в виді прикладних програм різного призначення.

Створення цих систем буде здійснюватися сучасним об'єктно-орієнтовним методом. Особливістю даної методології є програмний опис не процедур, що необхідні для рішення задач, а тих сутностей, які приймають участь в цих

процедурах і їх забезпечують. Тому що, процедури здійснюються, не абстрактним способом, а шляхом взаємодії деяких конкретних об'єктів, в відповідності з вже маючих в них властивостях. Якщо ці об'єкти і їх

властивості описані в програмі, то залишається тільки дати цим об'єктам

можливість взаємодіяти. Процедури, що необхідно виконати для рішення

задачі, буде здійснюватися як результат цієї взаємодії. Об'єктно-орієнтовне моделювання здійснюється таким шляхом послідовного виконання:

Об'єктна модель, яка є концептуальною базою об'єктно-орієнтовано

методології, має чотири головних елемента:

- абстрагування (виділення таких характеристик об'єкта, які відрізняють його від усіх інших видів об'єктів і таким чином чітко визначає особливості даного об'єкту з точки зору подальшого його аналізу);

- інкапсуляція (процес розділення елементів об'єкта, що визначає

його пристосування та поведінку. Інкапсуляція слугую для того, щоб ізолювати контрактні зобов'язання абстракцій від їх реалізації);

- модульність (властивість системи, що пов'язана з можливістю

декомпозиції на ряд зовнішніх пов'язаних, але слабо пов'язаних між собою

модулів);

- ієрархія (впорядкована система абстракція).

Без будь-якого з назначених вище елементів модель не буде об'єктно-орієнтованою.

Абстрагування і інкапсуляція доповнюють один одного: абстрагування

налаштоване на слідкування за поведінкою об'єкта, а інкапсуляція

займається внутрішнім пристосуванням. Це означає наявність двох частин в класі: інтерфейсу та реалізації. Інтерфейс відображає зовнішню поведінку,

що описує абстракцію поведінки всіх об'єктів даного класу. Внутрішня

реалізація описує представлення цієї абстракції і механізми досягнення

бажаної поведінки об'єкта. Розділення інтерфейсу та реалізації дозволяє

захистити об'єкти від деталей реалізації більш низького рівня. Інкапсуляція

дозволяє вносити в програму зміни, зберігши її надійність і мінімізувати витрати на цей процес.

Для реалізації об'єктно-орієнтованого підходу будемо використовувати уніфіковану мову моделювання – UML.

В першу чергу при створенні моделей в даних роботі, буде створена діаграма варіантів використання. Вона являє собою найбільш загальне представлення функціонального призначення системи. Загальні діаграми фактично не залежать від предмету моделювання і можуть застосовуватися в

будь-якому проєкті не озираючись на предметну область та область рішень.

Варіант використання описує послідовність дій, що виконується системою у відповідь на подію, що з'являється деяким зовнішнім об'єктом.

На діаграмі використання застосовуються два типи сутностей: варіант використання та дійові особи, між якими встановлені основні типи відношень:

Згідно з етапами рішення задачі щодо розроблення моделі поведінки людей, для створення потрібної комп'ютерної системи потрібно виділити наступні його структурні елементи:

- користувач (оператор), що має можливість вводити дані для розрахунку. Сповіщення людей при появі небезпеки та проведення інструктажу евакуації.

- модуль системи для розрахунку поведінки людей на основі створених конструктором даних та збору статичних даних о сесіях моделювання поведінки, аналіз отриманих відомостей, а також порівняння ефективності методів моделювання;

- модуль візуалізації результатів;

- датчик руху, що відстежує час входу та виходу, кожної людини.

Програмні елементи системи, що розробляється, пропонується розділити на декілька окремих розширюваних модулів, кожен з них має свій інтерфейс користувача. Цей підхід дає змогу проводити більш глибоке

тестування та робити перевірку на появу помилок на всіх рівнях моделювання поведінки людей в надзвичайній ситуації.

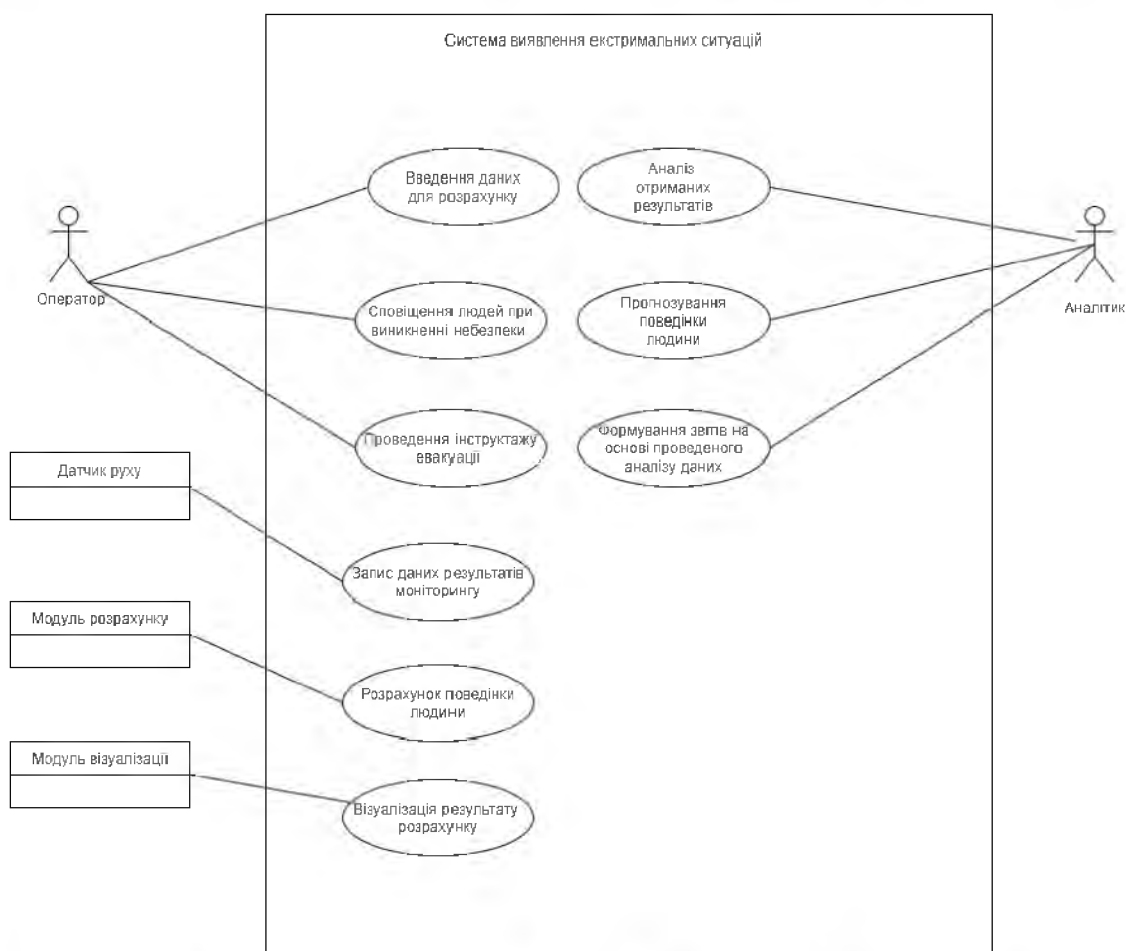


Рисунок 2.1 - Діаграма варіантів використання

Після запуску програми користувачеві потрібно авторизуватися в системі, ввести свій логін та пароль. Після цього у нього з'явиться доступ до введення початкових даних для моделювання поведінки. В свою чергу в введення даних входить: кількість людей, площа приміщення, швидкість натопту, ширина дверного отвору, кількість входів та виходів, щільність натопту. Потім конструктор виконує формалізацію вхідних параметрів, тобто зведення отриманих даних до конкретної форми. І вже модуль розрахунку проводить розрахунок поведінки людей по конкретній формулі. Наступним

кроком є імпортування отриманих даних, що були отримані в модуль візуалізації. Він аналізує ці дані і виводить їх користувачу. Кожен реалізований модуль є незалежним програмним продуктом, що взаємодіє з іншими модулями.

Діаграма послідовностей відображає виключно ті об'єкти, що безпосередньо беруть участь у взаємодії і не зображуються можливі статичні асоціації з іншими об'єктами. Для діаграми послідовності ключовим компонентом є динаміка взаємодії об'єктів у часі. Вона дає можливість поглянути на обмін повідомленнями

в часі, наочно відобразити послідовність процесу.

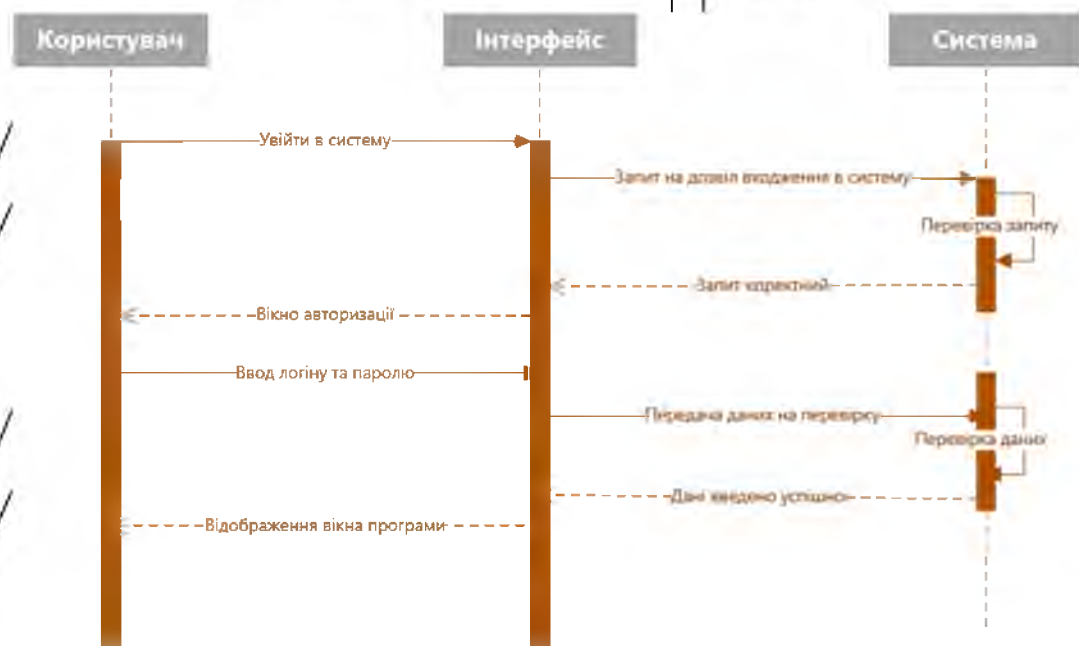


Рисунок 2.2 – Вхід у систему Інцидент ПІ – Вхід в систему.

Основний виконавець: користувач.

Передумова: на комп'ютері була запущена необхідна програма.

Головна послідовність – Рисунок 2.2;

– користувач натискає кнопку «Увійти»;

інтерфейс відправляє системі запит на дозвіл увійти до програми;
 система перевіряє правильність запиту, що був отриманий;
 система повертає до інтерфейсу відповідь про коректність запиту;

інтерфейс відображає вікно авторизації користувачу;
 користувач вводить логін та пароль у вікно;
 інтерфейс передає отримані дані системі на перевірку коректності;

система перевіряє правильність даних;
 система повертає інтерфейсу відповідь, що данні були введені успішно;
 інтерфейс відображає вікно програми.

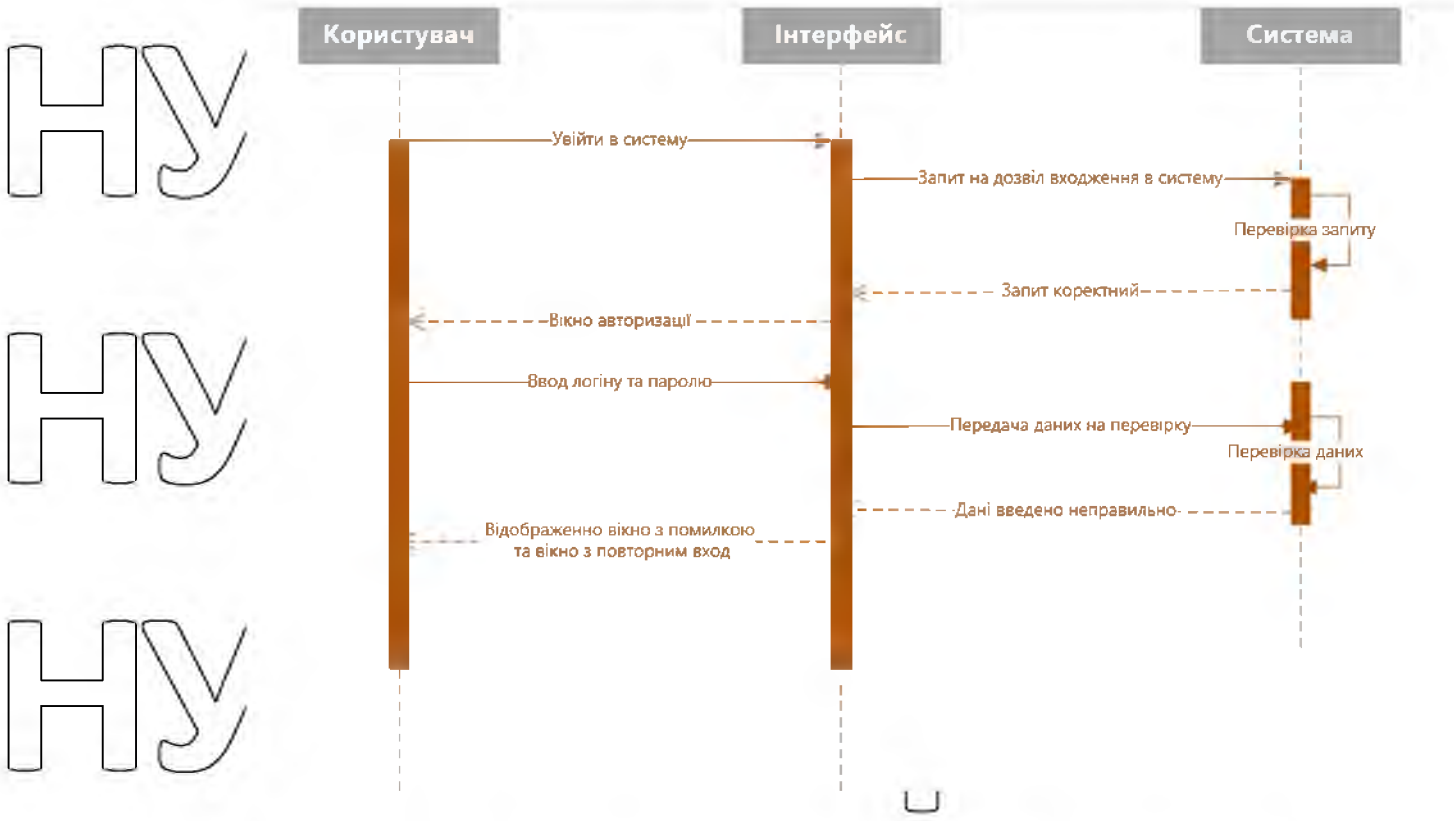


Рисунок 2.3 Невдалий вхід в систему Прецедент П1 Невдалий вхід в систему.

НУБІП України

Основний виконавець: користувач.

Передумова: на комп'ютері була запущена необхідна програма

Альтернативна невдала послідовність – Рисунок 2.3

1. користувач натискає кнопку «Увійти»;
2. інтерфейс відправляє системі запит на дозвіл увійти до програми;
3. система перевіряє правильність запиту, що був отриманий;
4. система повертає до інтерфейсу відповідь про коректність запиту;
5. інтерфейс відображає вікно авторизації користувачу;
6. користувач вводить логін та пароль у вікно;
7. інтерфейс передає отримані дані системі на перевірку коректності;
8. система перевіряє правильність даних;
9. система повертає інтерфейсу відповідь, що данні були введені некоректно;
10. інтерфейс відображає вікно про помилку та вікно для повторного входу.

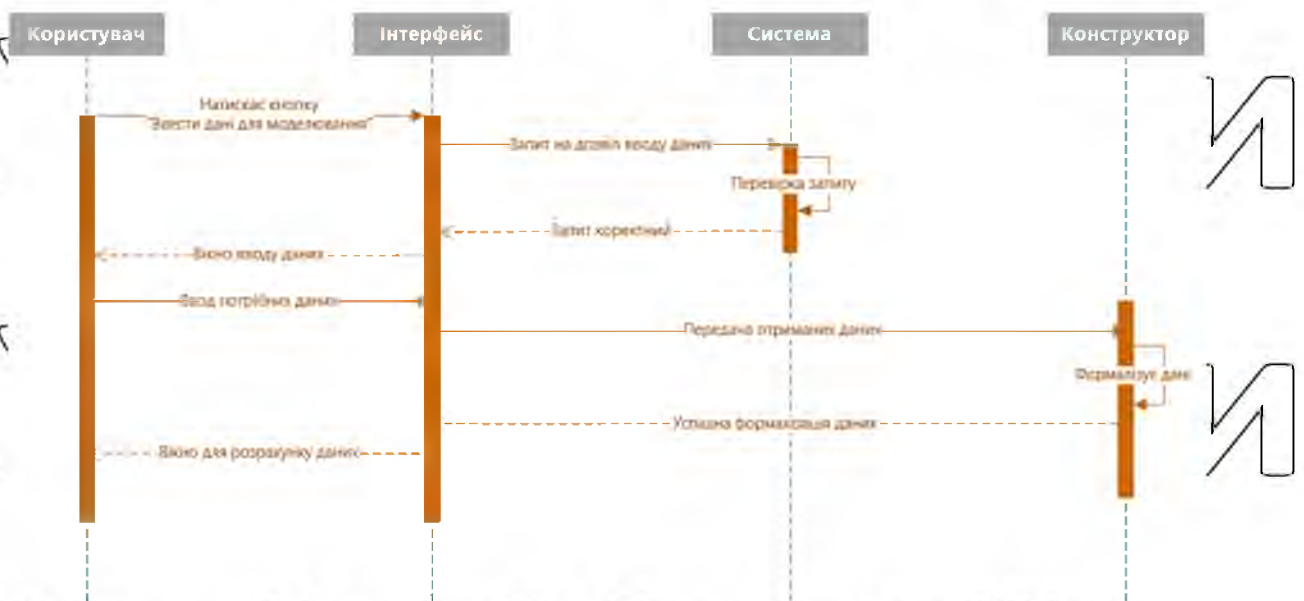


Рисунок 2.4) Введення даних для моделювання Преседент ПЗ – Введення даних для моделювання

НУБІП України

Основний виконавець: користувач.

Передумова: Користувач авторизувався у системі Головна послідовність. Рисунок 2.4.

1. користувач натискає кнопку «Ввести дані для моделювання»;
2. інтерфейс відправляє системі запит на дозвіл вводу даних;
3. система перевіряє правильність запиту, що був отриманий;
4. система повертає до інтерфейсу відповідь про коректність запиту;
5. інтерфейс відображає вікно вводу даних користувачу;
6. користувач вводить в форму потрібні дані;
7. інтерфейс передає конструктору отриману інформацію;
8. конструктор формалізує дані, що були введені;
9. конструктор повертає інтерфейсу відповідь про успішну формалізацію даних;
10. інтерфейс відображає вікно для розрахунку даних.

НУБІП України

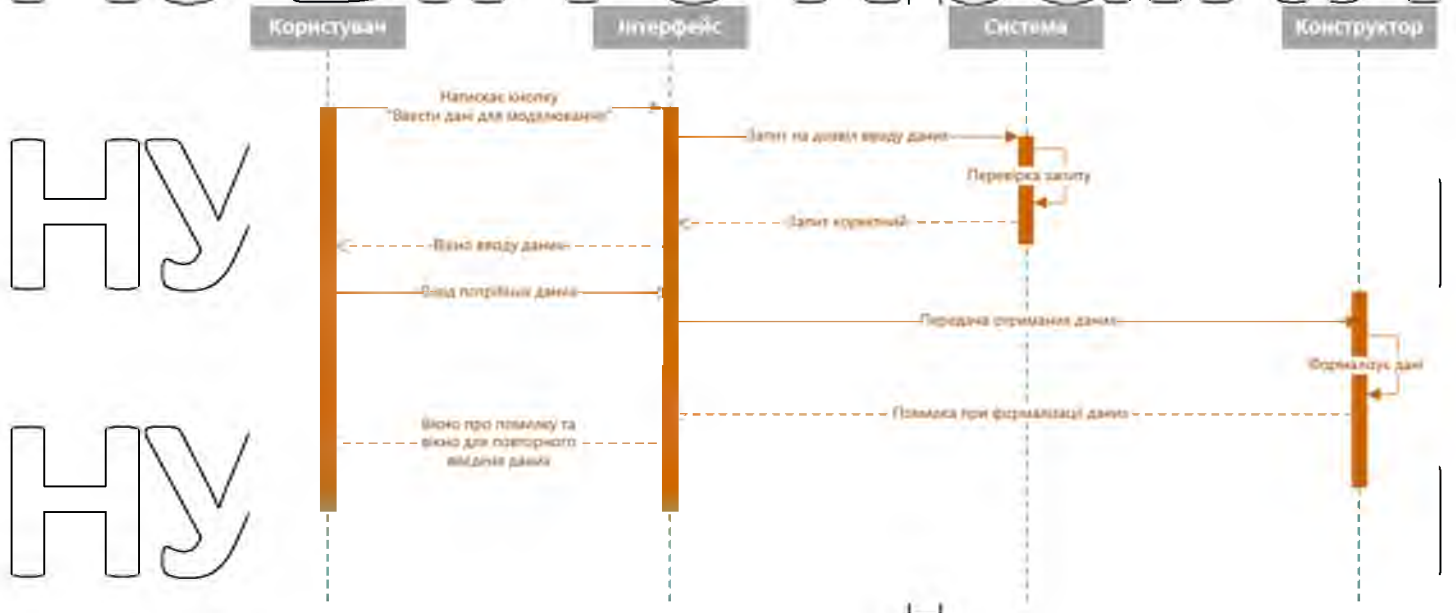


Рисунок 2.5 – Помилка введення даних для моделювання

Альтернативна послідовність

НУБІП України

1. користувач натискає кнопку «Ввести дані для моделювання».

2. інтерфейс відправляє системі запит на дозвіл вводу даних;
 3. система перевіряє правильність запиту, що був отриманий;
 4. система повертає до інтерфейсу відповідь про коректність запиту;

5. інтерфейс відображає вікно вводу даних користувачу;
 6. користувач вводить в форму потрібні дані;
 7. інтерфейс передає конструктору отриману інформацію;
 8. конструктор формалізує дані, що були введені;

9. конструктор повертає інтерфейсу відповідь про помилку і неможливість формалізувати дані;
 10. інтерфейс відображає вікно про помилку та вікно для повторного введення даних.

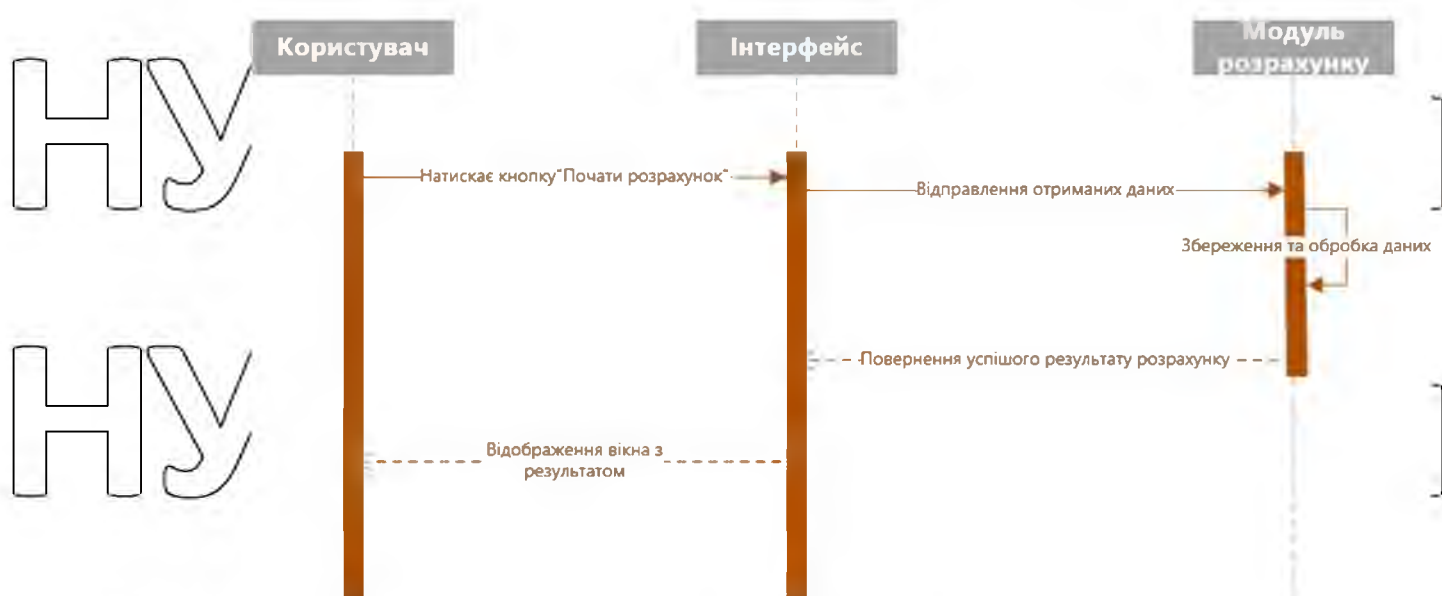


Рисунок 2.6 - Розрахунок даних Прецедент П4 – Розрахунок даних.
 Основний виконавець: користувач.

Передумова: Користувач ввів дані для моделювання. Головна послідовність: Рисунок 2.6.

1. користувач натискає кнопку «Почати розрахунок»;

2. інтерфейс відправляє отримані дані модулю розрахунку;
 3. модуль розрахунку зберігає та оброблює дані;
 4. модуль розрахунку повертає інтерфейсу успішно виконаний результат розрахунку;

5. інтерфейс відображає вікно з результатом.

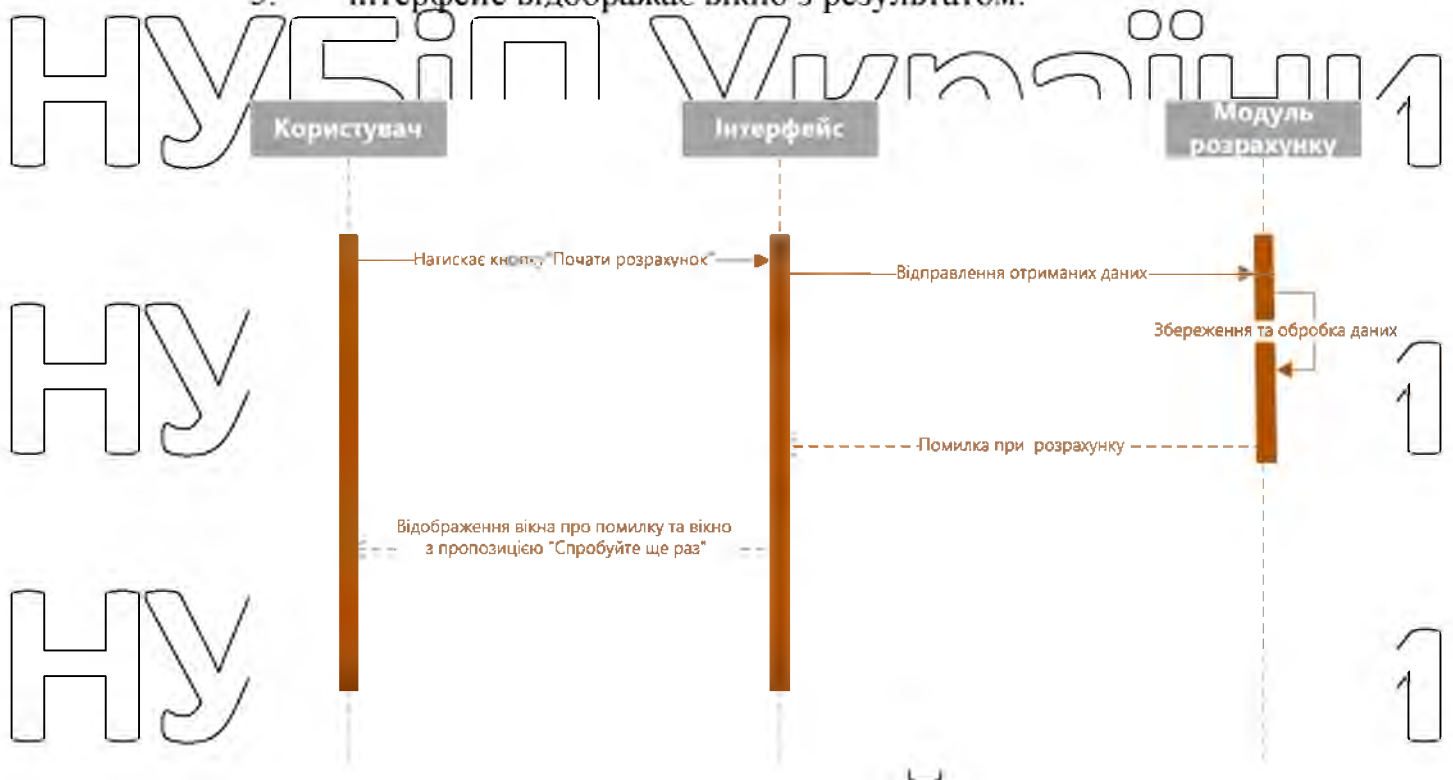


Рисунок 2.7 – Помилка при розрахунку даних

Альтернативна послідовність: Рисунок 2.7.

1. користувач натискає кнопку «Почати розрахунок»;
2. інтерфейс відправляє отримані дані модулю розрахунку;
3. модуль розрахунку зберігає та оброблює дані;
4. модуль розрахунку повертає інтерфейсу помилку при розрахунку даних;
5. інтерфейс відображає користувачу вікно з описом помилки та пропонує спробувати розрахувати ще раз.

Прецедент П5 – Візуалізація результату Основний виконавець: Користувач.

Передумова) Користувач розрахував дані для моделювання. Головна послідовність. Рисунок 2.8.

1. користувач натискає кнопку «Моделювати розрахунок»;
2. інтерфейс відправляє отриманні дані модулю візуалізації;
3. модуль візуалізації в свою чергу зберігає та оброблює дані, що надійшли;
4. модуль візуалізації повертає успішно змодельований результат;
5. інтерфейс відображає користувачу вікно з моделюванням

розрахунку.

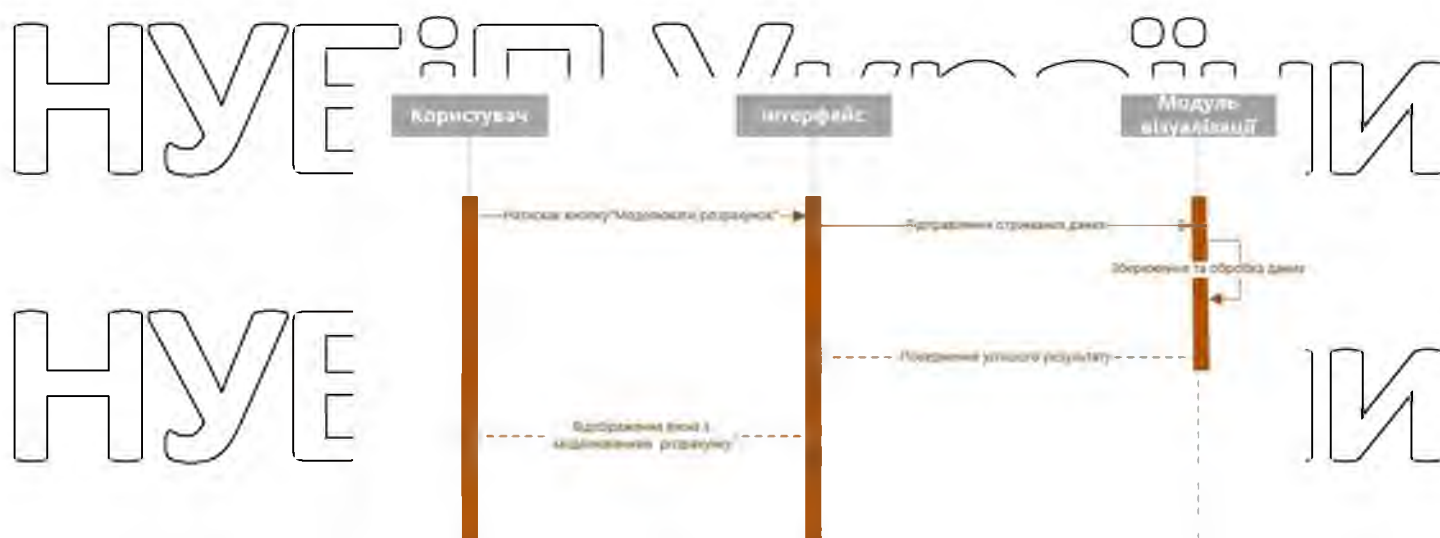


Рисунок 2.8 – Візуалізація результату. Альтернативна послідовність

Рисунок 2.9.

1. користувач натискає кнопку «Моделювання розрахунку»;
2. інтерфейс відправляє отриманні дані модулю візуалізації;
3. модуль візуалізації в свою чергу зберігає та оброблює дані, що надійшли;
4. модуль візуалізації повертає інтерфейсу помилку при моделюванні результату;
5. інтерфейс відображає користувачу вікно з описом помилки та пропонує спродувати ще раз.

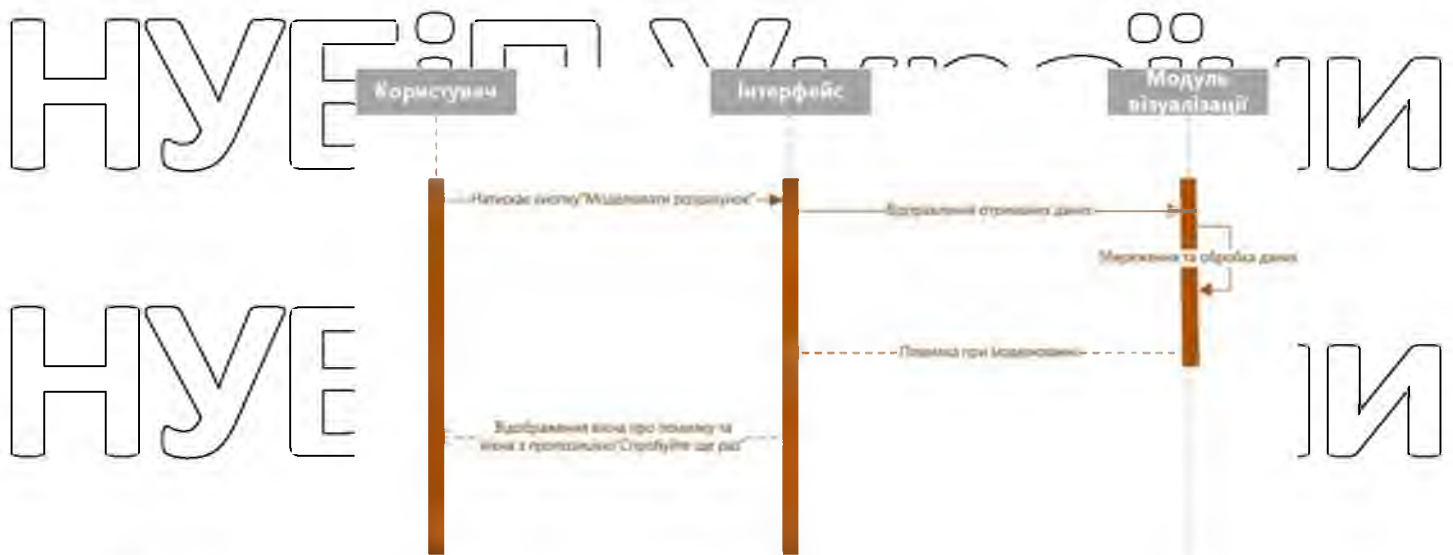


Рисунок 2.9- Помилка при візуалізації результату

Прецедент П6 – Отримання інформації про результати моделювання

Основний виконавець: Користувач.

Передумова. На комп'ютері попередньо була запущена програма.

Головна послідовність: Рисунок 2.10

1. користувач натискає кнопку «Увійти в систему»;
2. інтерфейс відправляє системі запит на дозвіл увійти до програми;
3. система перевіряє правильність запиту, що був отриманий;
4. система повертає до інтерфейсу відповідь про коректність запиту;
5. інтерфейс відображає вікно авторизації користувачу;
6. користувач вводить логін та пароль у вікно;
7. інтерфейс передає отримані дані системі на перевірку коректності;
8. система перевіряє правильність даних;
9. система повертає інтерфейсу відповідь, що данні були введені успішно;
10. інтерфейс відображає вікно програми;
11. користувач натискає на меню;

12. інтерфейс відображає користувачу відображає можливі варіанти з меню;

13. користувач натискає на довідку то обирає «Інформація про моделювання»;

14. інтерфейс відображає користувачу вікно з інформацією про моделювання.

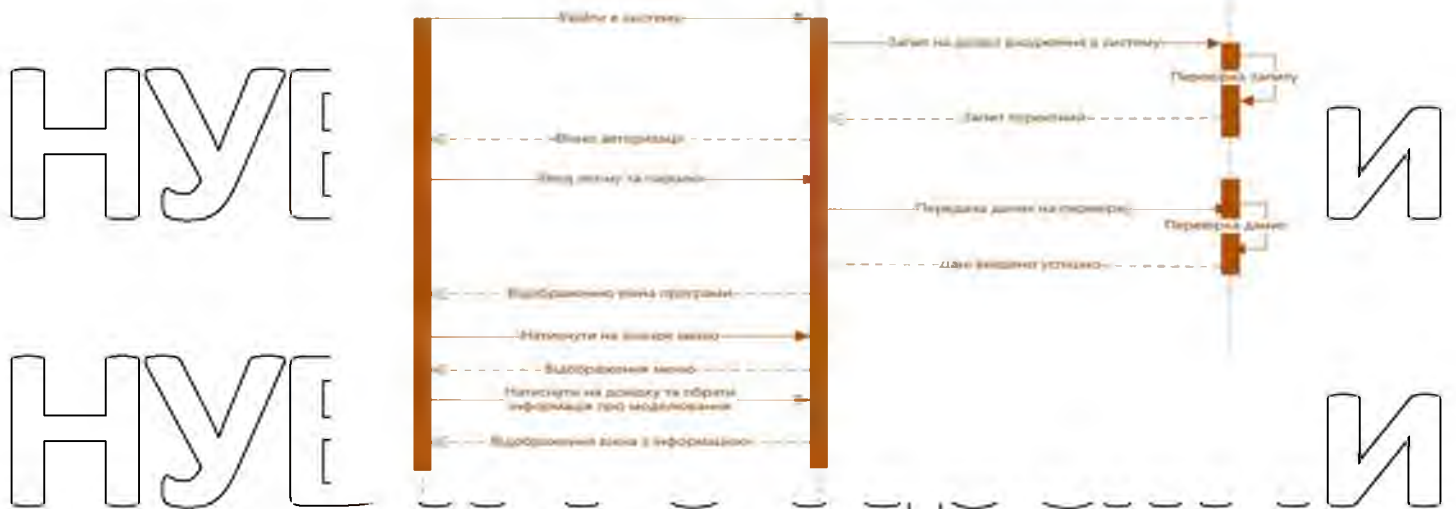


Рисунок 2.10 – Отримання інформації про результати моделювання

Важливе місце в об'єктно-орієнтованому проектуванні займає розробка логічної моделі системи у вигляді діаграми класів. Вона слугує для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого проектуванні. Діаграма класів може відображати різні взаємозв'язки між окремими суностями предметної області, такими як підсистеми та об'єкти, а ще описувати їх внутрішню структуру і типи відносин. Діаграма класів може вміщати в себе інтерфейси, пакети, відносини і окремі екземпляри, наприклад, об'єкти та зв'язки.

НУБІП України



Рисунок 2.11 – Діаграма класів

Графічний клас ілюструється у вигляді прямокутника. Він розділений частіше всього на три сектори: ім'я класу, атрибути та операції.

У класів-інтерфейсів може бути відсутні атрибути, як наприклад, у діаграмі, що зображена вище. Між класами встановлені відносини такі як: залежність, асоціація, узагальнення, реалізація.

Діаграму станів використовують для опису послідовностей переходів об'єкту із одного стану в інше. Вона покаже всі можливі стани, в яких може знаходитись об'єкт, а також процес зміни стану в результаті зовнішнього впливу. Загалом діаграма станів підходить для опису одної системи в рамках декількох прецедентів.



Рисунок 2.12 – Діаграма стану

Однак вона вважається неефективною при використанні для формалізації поведінки деяких взаємодіючих об'єктів. Основними елементами діаграми є стан та перехід. Характеристика стану систему слабо залежить від логічної структури, що зафіксована в діаграмі класів. Тому, враховуючи стан системи, нам слід на деякий час вилучити її зі структурних властивостей об'єкта і розглядати його у абсолютно різних категоріях, що формують динамічний контекст модельованої поведінки системи.

Діаграма розгортання призначена для відображення програмних елементів та компонентів, які знаходять лише під час виконання. У цьому випадку програма, яка представляє виконуваний файл або динамічні бібліотеки, представляє лише екземпляр компонента. Компоненти, які не використовуються під час руху, не показані на схемі розподілу. Тому компоненти з програмними джерелами можна знайти лише у діаграмах компонентів. Вони не вказані на діаграмі розгортання.

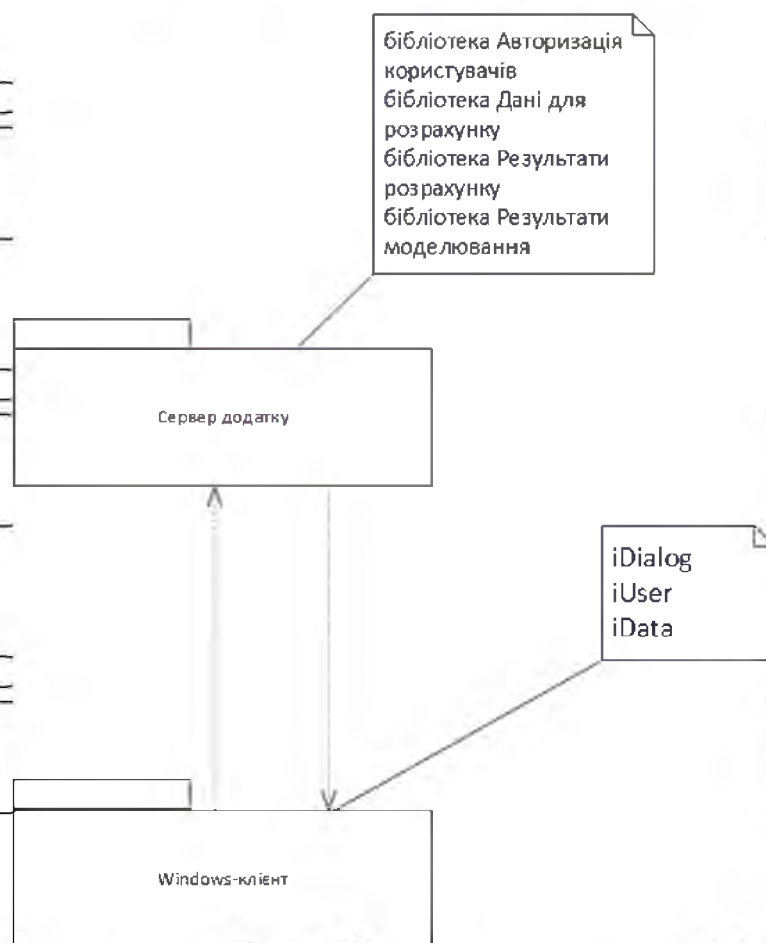


Рисунок 2.13 – Діаграма розгортання

Діаграма розгортання містить графічне зображення процесорів, пристроїв, процесів та їх взаємозв'язків. На відміну від логічних діаграм, схема розгортання для системи в цілому є однорідною, оскільки вона повинна повністю відображати можливості реалізації. Насправді ця діаграма

завершує процес для певної програмної системи, і її розробка зазвичай є завершальним етапом проектування моделі.

В системі моделювання поведінки людей в надзвичайних ситуаціях ми припустили, що користувач може самостійно реєструватися на сервері, тоді компонент інтерфейсу iDialog повинен бути розташовуватись на Windows-клієнт.

Бібліотека Авторизації користувача може бути розташована на сервері додатку, як і бібліотека Дані для розрахунку. Інтерфейс iDialog повинен після авторизації користувача надавати йому доступ до Введення даних.

Бібліотека Результати розрахунку розташовані на сервері додатку. Інтерфейс iUser звертається до цієї бібліотеки з Windows-клієнт, який використовує вже авторизований користувач для отримання результату після розрахунку введених даних. Також iUser може звертатись до бібліотеки

Результати моделювання, що розташована на Сервері додатку з метою отримання результатів моделювання, після того як отримала дані розрахунку і почала моделювання. Ще на Windows-клієнті розташовуватись інтерфейс iData, який може звертатися до Серверу додатку, щоб переглянути бібліотеки

Результати розрахунку, Результати моделювання які в собі зберігають дані попередніх користувань.

Покращений алгоритм ясно показує принцип функціонування моделі, приховуючи деталі конкретної реалізації і взаємодію окремих блоків схеми, що допомагає побачити загальний напрямок моделі.

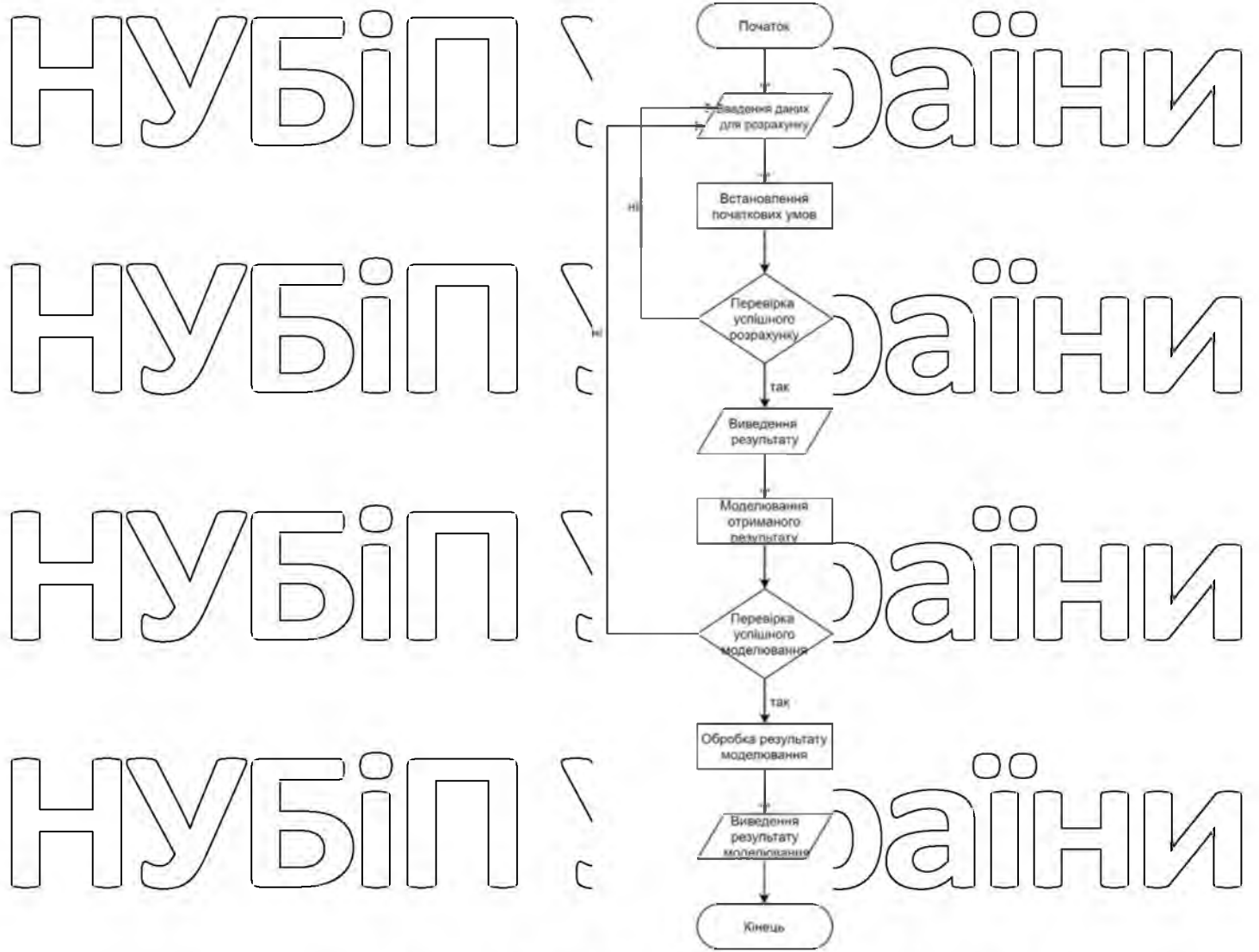


Рисунок 2.14 – Блок-схема алгоритму системи

НУБІП України

НУБІП України

НУБІП України

НУБІП України

3 РОЗРОБКА СИСТЕМИ

OLAP — це технологія, яка лежить в основі багатьох додатків Business

Intelligence. OLAP — є потужною технологією для виявлення даних, що включає можливості безмежного перегляду звітів, складних аналітичних розрахунків та прогнозного планування сценаріїв «що буде, якщо».

Знання є основою всіх успішних рішень. Для вдалого аналізу даних треба постійно планувати і звітують про операційну діяльність, щоб

максимізувати ефективність.

Аналітики скажуть вам, що чим більше вибіркового даних у вас є, тим більша ймовірність, що отримана статистика буде правдивою. Природно, чим більше даних про конкретну діяльність можна отримати, тим більша ймовірність того, що план покращення цієї діяльності буде ефективним.

Зібрати дані за допомогою багатьох різних систем, але тут постає запитання «Як зібрати всі дані разом, щоб створити точну, надійну та швидку інформацію про бізнес?».

Той хто точно та швидко може скористатися перевагами та перетворити їх у спільні знання, безсумнівно, матиме кращі позиції для прийняття успішних рішень. Технологія OLAP була визначена як здатність досягати «швидкого доступу до спільної багатовимірної інформації».

Враховуючи здатність технології OLAP створювати дуже швидкі агрегації та обчислення базових наборів даних, можна зрозуміти її корисність у допомозі приймати кращі та швидші «обгрунтовані» рішення.

Для розробки системи було побудована схема розробки системи:

НУБІП України



Рис 3.1 – Архітектура системи

На схемі зображено Оператор, який взаємодіє з Модулем введення інформації, що в свою чергу взаємодіє з базою даних системи. База даних — це набір пов'язаних даних, що представляють деякі елементи реального світу. В даній системі було вирішено також створити сховище даних, що є інформаційною системою, яка зберігає історичні та комутативні дані з одного чи кількох джерел. База даних призначена для запису даних, тоді як сховище даних призначене для аналізу даних. Сховище даних — це предметно-орієнтований збір даних. База даних використовує онлайнову обробку транзакцій (OLTP) тоді як сховище даних використовує онлайн-аналітичну обробку (OLAP). Таблиці та об'єднання бази даних є складними, оскільки вони нормалізовані, тоді як таблиці та об'єднання в сховищі даних прості, оскільки вони денормалізовані.

Далі наведено причини використання у системі виявлення екстремальних ситуацій бази даних та сховища:

Основні причини використання системи баз даних:

- забезпечує безпеку даних і доступ до них;
- база даних пропонує різноманітні методи зберігання та отримання даних;
- база даних діє як ефективний обробник, щоб збалансувати потреби кількох програм, що використовують одні й ті ж дані СУБД пропонує обмеження цілісності, щоб отримати високий рівень захисту, щоб запобігти доступу до заборонених даних;

– база даних дозволяє отримувати доступ до одночасних даних таким чином, що лише один користувач може отримати доступ до тих самих даних одночасно.

Важливі причини використання сховища даних:

– сховище даних допомагає бізнес-користувачам отримувати доступ до критичних даних із деяких джерел в одному місці;

– надає послідовну інформацію про різні міжфункціональні види діяльності;

– допомагає інтегрувати багато джерел даних, щоб зменшити навантаження на виробничу систему;

– сховище даних допомагає скоротити загальний час обробки для аналізу та звітності;

– сховище даних допомагає користувачам отримувати доступ до критичних даних з різних джерел в одному місці, тому заощаджує час користувача на отримання інформації з кількох джерел. Ви також можете легко отримати доступ до даних із хмари.

– сховище даних дозволяє зберігати велику кількість історичних даних для аналізу різних періодів і тенденцій, щоб робити прогнози на майбутнє.

– відокремлює аналітичну обробку від транзакційних баз даних, покращуючи продуктивність обох систем;

Ще одним модулем в системі є «Модуль аналітики» з яким співпрацює Аналітик. Зовнішніми джерелами, є «Модуль розрахунку» та дані з датчику.

Перед створенням будь-якої бази даних треба запланувати, що вона повинна включати і як різні аспекти працюють разом. Таке планування гарантує, що база даних має необхідний дизайн для використання за призначенням. Потім можна використовувати схему для реалізації дизайну бази даних.

Схеми баз даних — це креслення, які допомагають розробникам уявити, як мають бути створені бази даних. Вони забезпечують орієнтир, який вказує, які поля інформації містить проєкт. Якщо є якісь проблеми,

розробник може просто звернутися до схеми. Також використовують схеми, щоб переконатися, що база даних повністю розроблена перед впровадженням. Це економить дорогоцінний час і кошти, оскільки після впровадження бази даних внесення змін може бути складним. Схема

гарантує, що всі зацікавлені сторони повністю розглянули кожен аспект проекту, що зменшує необхідність змін.

Оскільки схема зірка є адаптацією моделі реляційної бази даних, схема сніжинка — це адаптація схеми зірка. Його назва походить від того, як можна було б зобразити ERD (діаграму зв'язків сутності) схеми сніжинки: ви

здогадалися, вона починає виглядати як сніжинка. Як і зі схемою зірка, схема сніжинка має центральну таблицю фактів, яка зберігає основні точки даних і посилання на її розмірні таблиці. На відміну від зіркової схеми, таблиці розмірів схеми сніжинки можуть мати власні таблиці розмірів, таким чином розширюючи, наскільки описовим може бути розмір.

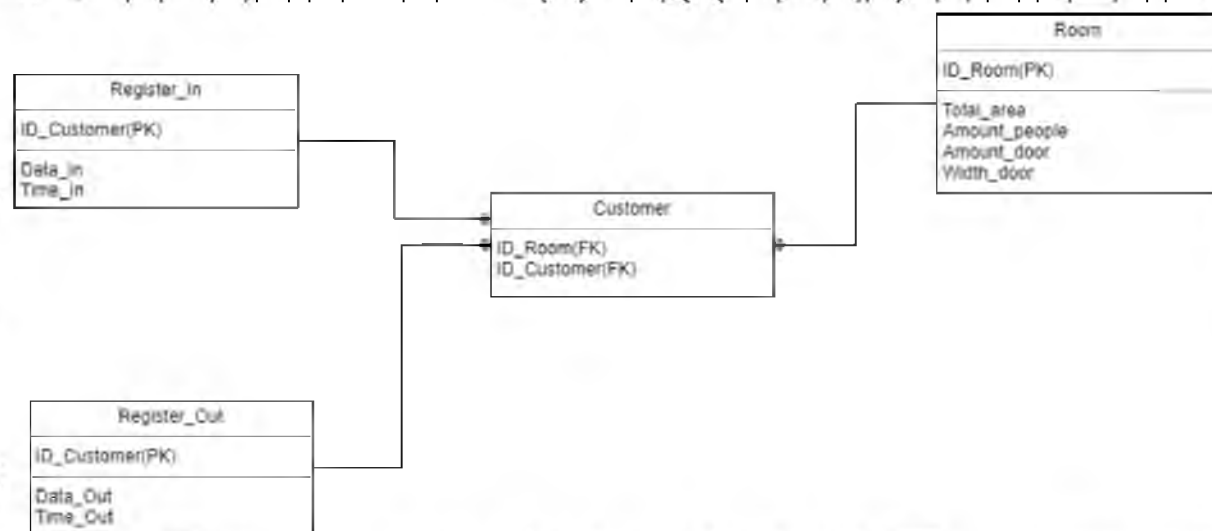


Рис 3.2) Структура бази даних

Таблиця Register_In містить атрибути ID_Customer, Data_In, Time_In.

Атрибут ID_Customer зв'язує таблицю Register_In з таблицею Customer.

Атрибут ID_Customer, використовується для того щоб надати детальну інформацію про кожного відвідувача. Атрибути Data_In та Time_In фіксують

дату та час приходу відвідувачів. Таблиця Register_Out містить атрибути ID_Customer, Data_Out, Time_Out. Атрибут ID_Customer зв'язує таблицю Register_In з таблицею Customer. Атрибути Data_Out і Time_Out фіксують дату та час, коли відвідувач покинув приміщення.

Таблиця Customer містить атрибути ID_Customer, ID_Room. Атрибут ID_Room використовується для того щоб надати детальну інформацію про характеристики приміщення, аудиторій, зал, кабінетів тощо. Таблиця Room містить атрибути ID_Room, Total_area, Amount_people, Amount_door, Width_door. Атрибут ID_Room зв'язує таблицю Room з таблицею Customer.

Атрибут Total_area фіксує загальну площу кожного приміщення. Amount_people фіксує кількість людей в приміщенні. Amount_door фіксує кількість дверей. Атрибут Width_door зберігає інформацію про ширину дверного отвору.

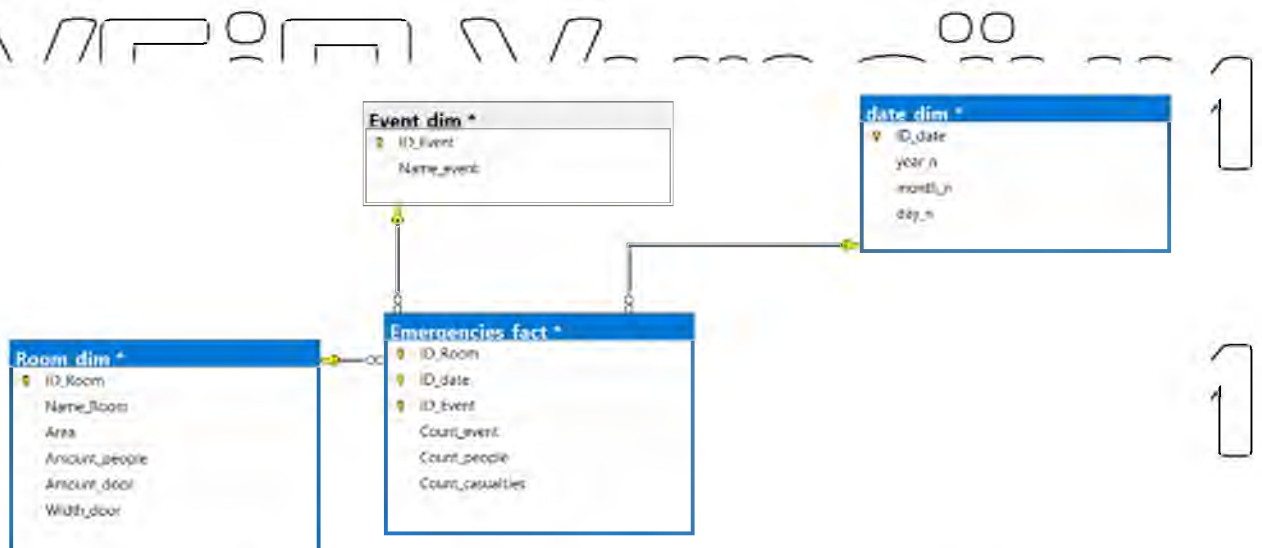


Рис 3.3 – Структура сховища даних

Наступні концепції підкреслюють деякі з найкращих ідей та принципів проектування, які використовуються для створення традиційних сховищ даних.

Традиційна архітектура сховища даних часто має трирівневу структуру, що складається з наступних рівнів:

Нижній рівень. Цей рівень містить сервер бази даних, який використовується для отримання даних з різних джерел, таких як: транзакційні бази даних, що використовуються для інтерфейсних додатків.

Проміжний рівень. Проміжний рівень містить сервер OLAP, який перетворює дані на структуру, більш підходящу для аналізу та складних запитів. OLAP Server може використовуватися двома способами: або як розширена система управління реляційною базою даних, яка зіставляє операції з багатовимірними даними зі стандартними Relational OLAP, або як багатовимірна модель OLAP, яка безпосередньо реалізує дані та багатовимірні операції.

Верхній рівень. Верхній рівень – рівень клієнта. Цей рівень містить інструменти для загального аналізу даних, звітності та аналізу даних.

У традиційній архітектурі є три популярні моделі сховища даних:

віртуальне, корпоративна пам'ять та вітрина даних:

Віртуальне сховище даних - це окремий набір баз даних, які можуть використовуватися спільно, щоб користувач міг ефективно отримувати доступ до всіх даних, як якщо вони зберігалися в єдиному сховищі даних;

Модель відображення даних використовується для створення звітів та аналізу конкретних сфер бізнесу. Ця модель інвентаризації збирає дані з різних вихідних систем, пов'язаних із певною сферою бізнесу;

Модель корпоративного сховища даних забезпечує повне сховище даних, що охоплює всю організацію. Ця модель передбачає, що сховище даних є ядром корпоративної IT-системи з інтегрованими даними всім бізнес-одиниць.

Діаграми у вигляді зірочок та сніжинок – це два способи структурувати сховище даних.

Перша має центральне сховище даних, що зберігається у таблиці фактів.

Діаграма поділяє таблицю фактів на ряд денормалізованих таблиць вимірювань. Таблиця фактів містить агреговані дані, які використовуються для звітів, а таблиця вимірів описує збережені дані.

Денормалізовані проекти менш складні, оскільки дані згруповані. У таблиці фактів використовується одне посилання додавання кожної таблиці вимірів. Простіший дизайн зірчастої діаграми значно спрощує написання складних запитів.

А от діаграма "сніжинка" відрізняється тим, що використовує стандартизовані дані. Нормалізація означає ефективну організацію даних, щоб усі залежності даних були визначені, а кожна таблиця мала мінімальну надмірність. Таким чином, окремі таблиці вимірів поділені на окремі таблиці вимірів.

Схема Flake використовує менше місця для зберігання та краще зберігає цілісність даних. Основним недоліком є складність запитів, необхідних доступу до даних - кожен запит має проходити кілька таблиць злиття, щоб отримати правильні дані.

MapReduce — це модель програмування або шаблон у рамках Hadoop, який використовується для доступу до великих даних, що зберігаються у файловій системі Hadoop (HDFS). Це основний компонент, невід'ємний елемент функціонування фреймворку Hadoop.

Це полегшує одночасну обробку, розбиваючи петабайти даних на менші фрагменти та паралельно обробляючи їх на серверах Hadoop. Загалом, він об'єднує всі дані з кількох серверів, щоб повернути консолідований вихід до програми.

MapReduce колись був єдиним методом, за допомогою якого можна було отримати дані, збережені в HDFS, але тепер це не так. Сьогодні існують інші системи на основі запитів, такі як Hive і Pig, які використовуються для отримання даних з HDFS за допомогою операторів, подібних до SQL. Однак вони зазвичай виконуються разом із завданнями, написаними за допомогою моделі MapReduce. Це тому, що MapReduce має унікальні переваги.

В основі MapReduce є дві функції: Map і Reduce. Вони послідовні один за одним.

Функція `Map` приймає вхідні дані з диска як пари <ключ,значення>, обробляє їх і створює інший набір проміжних пар <ключ,значення> як вихід.

Функція `Reduce` також приймає вхідні дані як пари <ключ,значення> і створює пари <ключ,значення> як вихідні дані.

Типи ключів і значень відрізняються залежно від варіанта використання. Усі входи та виходи зберігаються в HDFS. Хоча карта є обов'язковим кроком для фільтрації та сортування вихідних даних, функція зменшення не є обов'язковою.

`<k1, v1> -> Map() -> list(<k2, v2>)`

`<k2, list(v2)> -> Reduce() -> list(<k3, v3>)`

`Mappers` і `Reducers` — це сервери Hadoop, які запускають функції

Лістинг 4.1 – Функція Map Reduce

```
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(ExceptionCount.class);
    conf.setJobName("exceptioncount");
```

```
    conf.setOutputKeyClass(Text.class);
```

```
    conf.setOutputValueClass(IntWritable.class);
```

```
    conf.setMapperClass(Map.class);
```

```
    conf.setReducerClass(Reduce.class);
```

```
    conf.setCombinerClass(Reduce.class);
```

```
    conf.setInputFormat(TextInputFormat.class);
```

```
    conf.setOutputFormat(TextOutputFormat.class);
```

```
    FileInputFormat.setInputPaths(conf, new Path(args[0]));
```

```
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));
```

```
    JobClient.runJob(conf);
```

```
    }
```

Параметри — ім'я класу `MapReduce`, класи `Map`, `Reducer` і `Combiner`, типи введення і виведення, шляхи вхідних і вихідних файлів — визначаються в головній функції. Клас `Mapper` розширює `MapReduceBase` і реалізує інтерфейс `Mapper`. Клас `Reducer` розширює `MapReduceBase` і реалізує інтерфейс `Reducer`.

Технології великих даних для аналітики. Нині обсяг даних, якими зараз керують наші системи зберігання, перевищив потужність обробки традиційних систем, і це також стосується інтелекту даних. Розподілені обчислення широко використовувалися експертами та практиками до появи `BigData` для покращення послідовних рішень у даних середнього розміру.

Тим не менш, для більшості сучасних масштабних проблем розподілений підхід сьогодні стає обов'язковим, оскільки жодна пакетна архітектура не здатна вирішити такі масштаби. Крім рішень високопродуктивних обчислень, нові великомасштабні платформи обробки призначені для наближення розподіленої обробки до практиків і експертів, приховуючи технічні нюанси, похідні від цих середовищ. Для створення та підтримки цих платформ необхідні нові та складні проекти, що узагальнює використання розподілених обчислень для стандартних користувачів. У результаті швидкого розвитку середовища великих даних з'явилася безліч інструментів, парадигм і методів для вирішення різних випадків використання в промисловості та науці. Однак через таку велику кількість альтернатив практикам та експертам часто важко проаналізувати та вибрати.

Основною ідеєю `Hadoop` було створення загальної структури, яка може обробляти великомасштабні дані на кластері обладнання, без великих витрат на розробку правильний інструмент для кожної мети. Але нині `Hadoop` — це більше ніж одна технологія та повний програмний стек та екосистема, що складається з кількох компонентів верхнього рівня, які відповідають різним цілям.

Чотири модулі включають основний фреймворк `Hadoop` і працюють разом, щоб сформувати екосистему `Hadoop`:

Розподілена файлова система Hadoop (HDFS), як основний компонент екосистеми Hadoop. HDFS є розподіленою файловою системою, яка забезпечує високошвидкісний доступ до даних програми без необхідності попереднього визначення схем.

Ще один узгоджувач ресурсів (YARN): YARN — це платформа для управління ресурсами, відповідальна за керування обчислювальними ресурсами в кластерах та використання їх для планування програм користувачів. Він виконує планування та розподіл ресурсів у системі Hadoop.

MapReduce: MapReduce — це модель програмування для великомасштабної обробки даних. Використовуючи алгоритми розподілених і паралельних обчислень, MapReduce дає можливість перенести логіку обробки та допомагає писати програми, які перетворюють великі набори даних в один керований набір.

Hadoop Common: Hadoop Common включає бібліотеки та утиліти, які використовуються та використовуються іншими модулями Hadoop.

У ході дослідження буде використано служби аналізу SQL Server (SSAS) — це багатовимірний OLAP-сервер, а також механізм аналітики, який дозволяє розділяти і розділяти великі обсяги даних. Він є частиною Microsoft SQL Server і допомагає виконувати аналіз за допомогою різних вимірів. Він має 2 варіанти багатовимірного та таблицьного. Повною формою SSAS є SQL Server Analysis Services.

Архітектурний вигляд служб аналізу сервера SQL базується на тривірневій архітектурі, яка складається з

СУБД: дані з різних джерел, таких як Excel, база даних, текст, інші можуть бути залучені за допомогою інструменту ETL в СУБД.

SSAS: Зведені дані з СУБД поміщаються в кубу SSAS за допомогою проектів служб аналізу. Куби SSAS створюють базу даних аналізу, і коли база даних аналізу буде готова, її можна використовувати для багатьох цілей.

Клієнт: клієнти можуть отримати доступ до даних за допомогою інформаційних панелей, систем показників, порталів тощо

Куб є основною одиницею зберігання. Це набір даних, який був узагальнений, щоб дозволити запитам швидко повертати дані.

Використовуючи служби SSAS, було побудовано розгорнутий куб для свого проекту.

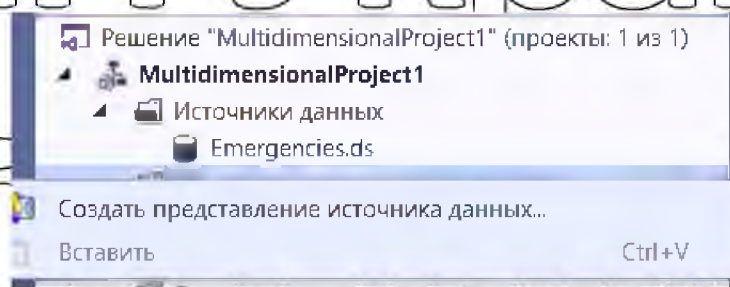


Рис. 3.4 – Створення джерела даних

Наступним кроком є створення представлення джерела даних

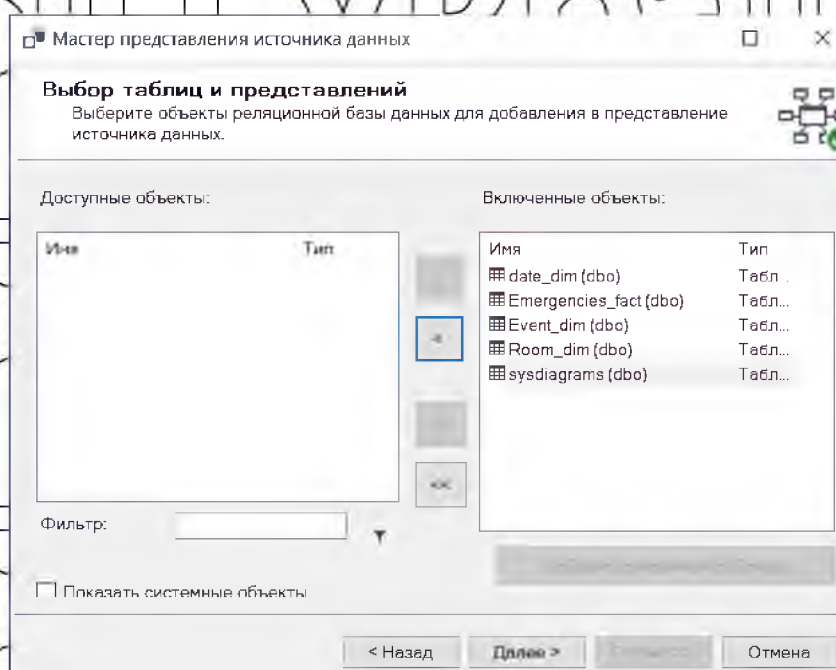


Рис. 3.5 – Представлення джерела даних

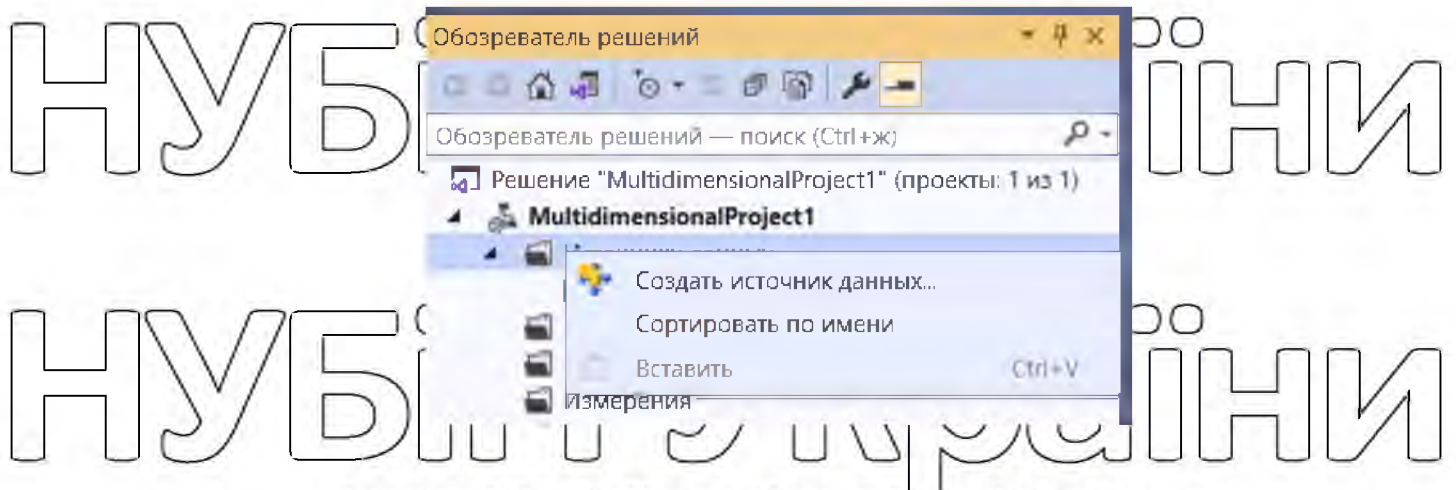


Рисунок 3.6 – Выбор нужных таблиц и представлений

И последний шаг, создания куба.

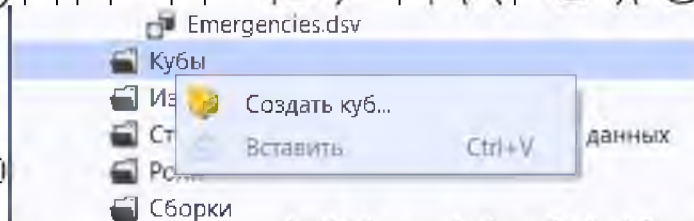


Рис. 3.7 – Создание куба

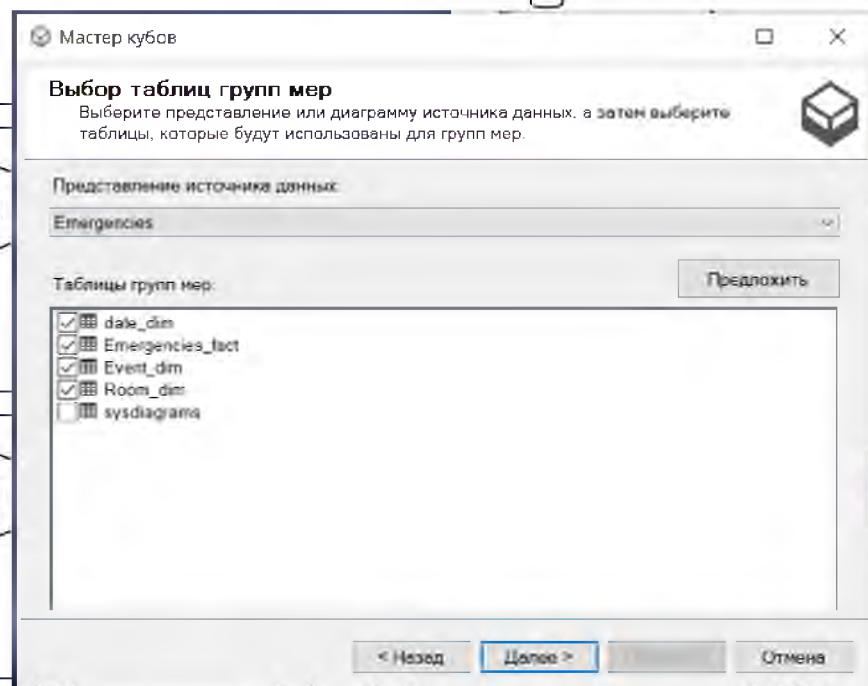


Рис. 3.8 – Выбор таблиц групп мер

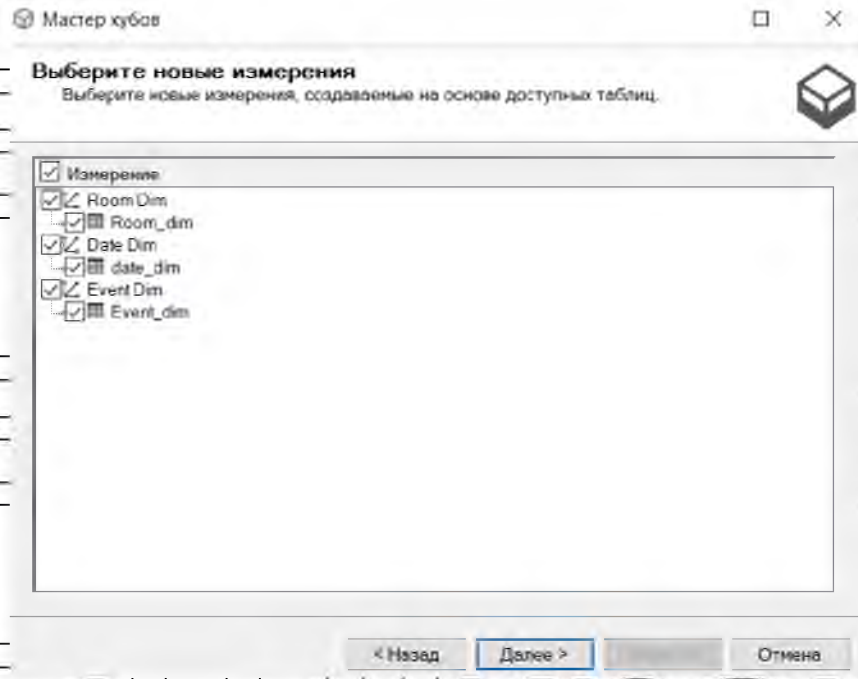


Рис. 3.9 – Вибір вимірів

Після створення кубу, я використовуючи службу SSIS, на основі процесів Data Flow, заповнила даними побудований куб. Далі буде зображено результат, який було отримано:



Рис. 3.10 – Потік управління

Задача потока данных: Поток вимірів



Рис. 3.11 – Потік вимірів



Рис. 3.12 – Потік фактів

Була встановлена відповідність між таблицею Emergencies_Fact заповненого даними сховища даних та порожнім сховищем з таблицею Emergencies.

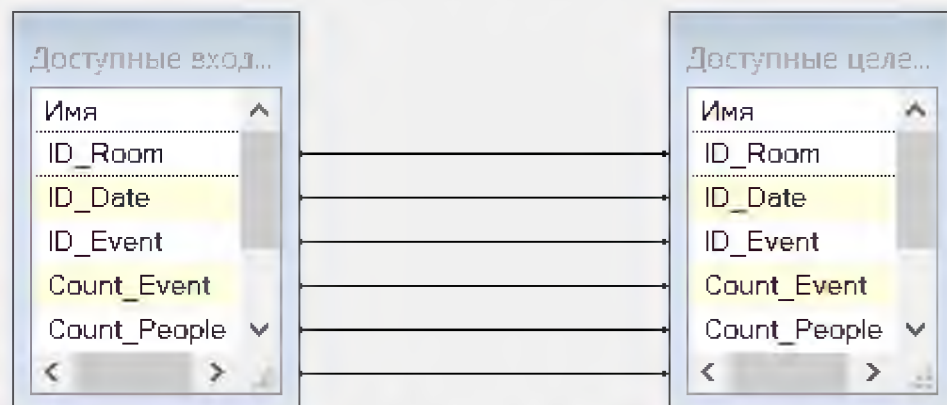


Рисунок 3.13 – Відповідність Emergencies_Fact

НУБІП України

Також встановлена відповідність між таблицею Date_dim заповненого даними сховища даних та порожнім сховищем с таблицею Date.

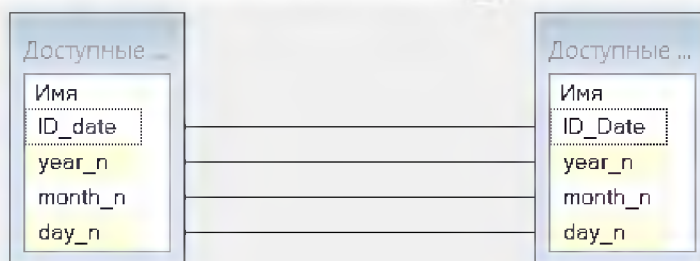


Рис 3.14 – Відповідність Date_dim

НУБІП України

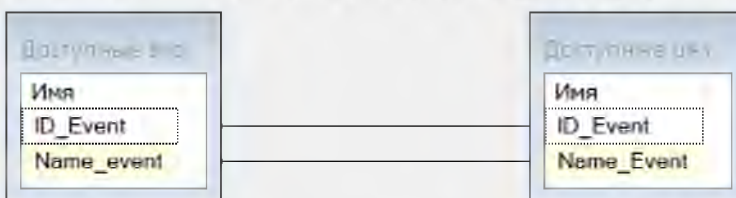


Рис. 3.15 – Відповідність Event_dim

НУБІП України



Рис. 3.16 – Відповідність Room_Dim

Міра являє собою стовпець, який містить дані, які підлягають

кількісному виміру, зазвичай числові, які можна агрегувати. Група заходів, що мають однакову деталізацію, утворює групу заходів.

Усі виміри, додані в папку «Вимір», можна використовувати в екземплярі куба. Атрибути, які додаються та обробляються на рівні виміру,

НУБІП України

відображаються у вимірі куба. Розміри, доступні в DSV, витягуються лише в структуру куба. Виміри куба можна редагувати, щоб додати нові атрибути або ієрархії.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

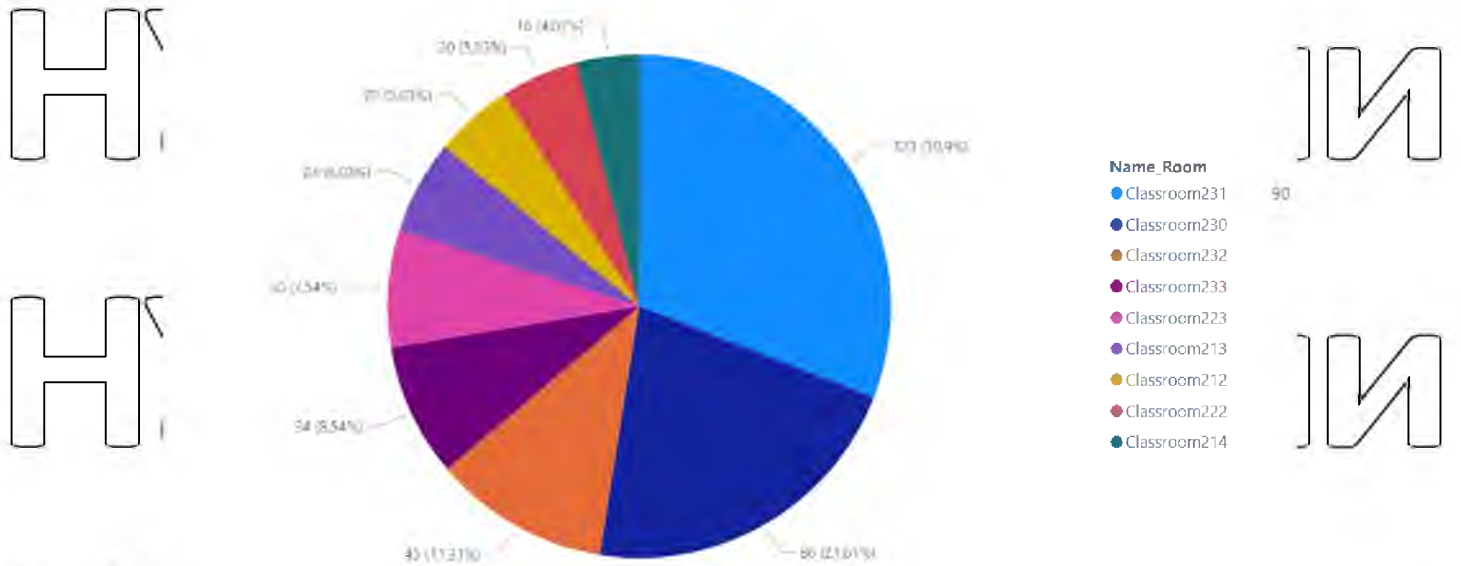
НУБІП України

НУБІП України

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

НУБІП України

Назад к отчету | COUNT_PEOPLE ПО NAME_ROOM



НУБІП України

НУБІП України

НУБІП України

Рис. 4.1 - Кругова діаграма кількості людей в аудиторіях

НУБІП України

У результаті аналізу цього звіту можна зробити висновок, що більшість людей, які приходять до корпусу в цей місяць, знаходяться в 231 та 230 аудиторіях. Так як місткість та площа цих аудиторій більша ніж в інших.

Тому і кількість людей тут найбільша. З цього випливає, що ці аудиторіях більше натовпу і в разі екстремальної ситуації більше вірогідність постраждалих.

НУБІП України

Name_Event	Classroom212	Classroom213	Classroom214	Classroom222	Classroom223	Classroom230	Classroom231	Classroom232	Classroom233	Всього
Вибух						0			0	0
Затоплення		1								1
Землетрус						0				0
Пожежа	0		0				1	1		2
Теракт				0						0
Всього	0	1	0	0	0	0	1	1	0	3

Рис. 4.2 - Матриця. Кількість екстремних ситуацій в заданих аудиторіях.

На даному звіті видно в яких аудиторіях траплялись зазначенні екстремальні події і в якій кількості. Дані введені за один місяць. У результаті аналізу цього звіту можна зробити висновок, що відбулось дві екстремальні події (пожежа та затоплення). Затоплення один раз в 212 та два рази пожежа в 231 та 232. Всього за місяць відбулось три екстремальні події.

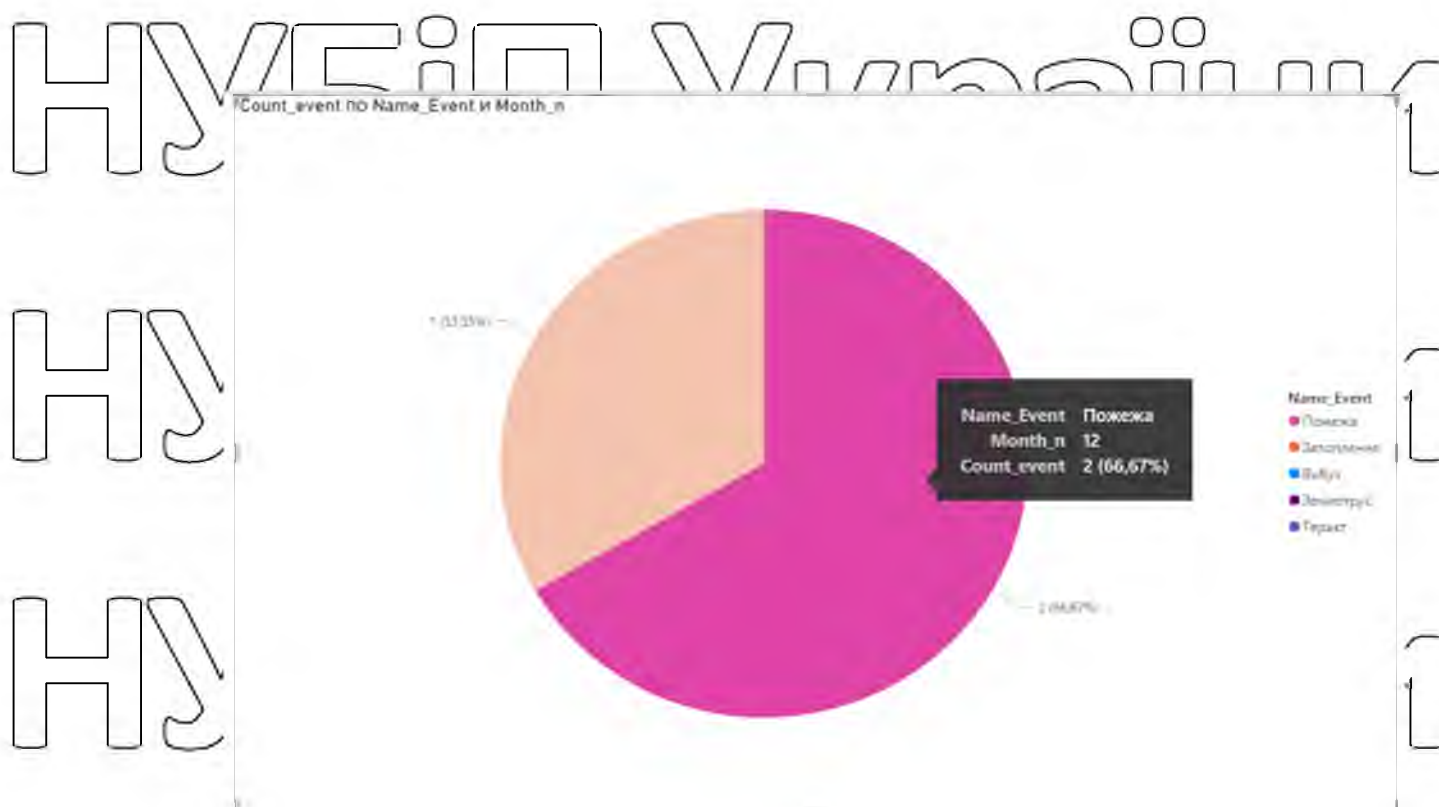


Рис.4.3 - Кругова діаграма видів екстремальних подій за місяць

На даному звіті ми бачимо, які види екстремальних ситуацій відбулись і який відсоток від всіх подій вони мають. Наприклад, пожежа займає 66,7% від усіх подій за місяць, а отоплення 33,3%

Для обчислення ключових показників ефективності (KPI), я використовувала програму для бізнес-аналітики від Microsoft Power BI.

За ціль я взяла середнє значення місткості аудиторії загалом. Як можна побачити по результатам фактичне значення значно менше цільового, тобто наповненість аудиторії є неповною. Результат будемо вважати позитивним, так як немає великої скупченості людей.

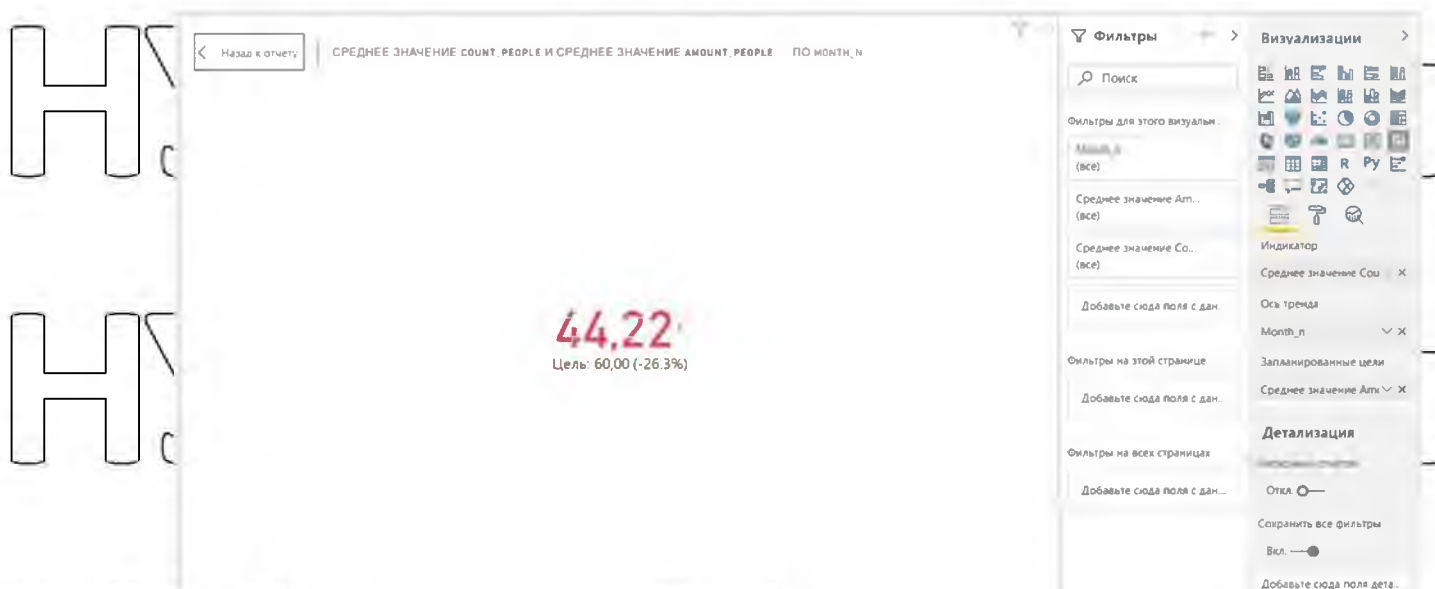


Рис.4.4 - Розрахунок KPI

ВИСНОВОК

Застосування OLAP-технологій дозволило ефективно проаналізувати різні екстремальні ситуації. Оцінити кількість та частоту їх виникнення з метою дослідження та раннього попередження. Завдяки методам Data Mining було прогнозовано вірогідність виявлення надзвичайної ситуації в певних місцях. У ході також було визначено шлях удосконалення дорадчої системи

Також було розроблено сховище даних для системи виявлення екстремальних ситуацій. На основі даних сховища здійснювались різні види аналізу і формування звітності. Використовуючи служби SSAS, було

побудовано розгорнутий куб для цього проекту та використовуючи службу SSIS, на основі процесів Data Flow, було заповнено даними побудований куб. На наступному етапі, був проведений аналіз даних, в результаті якого

за допомогою Power BI обчислили ключовий показник ефективності та

підготували звіти. Звіти дозволили в більш зручному для сприйняття вигляді, узагальнити й проаналізувати дані

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

НУБІП України

1. Науково-технічний журнал «Техногенно-екологічна

безпека». 2017. №1 Азаров С.І [Електронний ресурс]

Режим доступу:

<https://nuczu.edu.ua/sciencearchive/JournalTechnogenicAndEcologicalSafety/Vol-1/AzarovS.-Vol-1-2017-3-7.pdf>

НУБІП України

2. Системи моделювання і візуалізації процесу

евакуації працівників підприємства в надзвичайних

ситуаціях Савицька Е.О., Горбатов А.В. [Електронний ресурс]

Режим доступу: [https://cyberleninka.ru/article/n/sistema-modelirovaniya-i-](https://cyberleninka.ru/article/n/sistema-modelirovaniya-i-vizualizatsii-protsesta-evakuatsii-sotrudnikov-predpriyatiya-v-chrezvychaynyh-situatsiyah/viewer)

[vizualizatsii-protsesta-evakuatsii-sotrudnikov-predpriyatiya-v-chrezvychaynyh-situatsiyah/viewer](https://cyberleninka.ru/article/n/sistema-modelirovaniya-i-vizualizatsii-protsesta-evakuatsii-sotrudnikov-predpriyatiya-v-chrezvychaynyh-situatsiyah/viewer)

НУБІП України

3. Моделювання поведінки панікуючого наговпу в багаторівневому

розгалуженому приміщенні [Електронний ресурс] – Режим доступу: [http://crm-](http://crm-en.ics.org.ru/uploads/crmissues/crm_2013_3/13313.pdf)

[en.ics.org.ru/uploads/crmissues/crm_2013_3/13313.pdf](http://crm-en.ics.org.ru/uploads/crmissues/crm_2013_3/13313.pdf)

НУБІП України

4. Керування життєвим циклом програми інформаційних систем

[Електронний ресурс] – Режим доступу:

https://studme.org/184178/informatika/microsoft_visio

НУБІП України

5. A mathematical model of the pressure of an extreme ideology

on a society/ Francisco J. Santonja, Ana C. Tarazona, Rafael J. Villanueva.

ScienceDirect 2008.

6. Human Behavior Understanding in Big Multimedia Data Using CNN based

Facial Expression Recognition/ Muhammad Sajjad, Sana Zahir, Amin Ullah,

Zahid Akhtar, Khan Muhammad. *Springer Science+Business Media, LLC,*

part of Springer Nature 2019.

НУБІП України

7. Математична модель запобігання надзвичайним ситуаціям

терористичного характеру шляхом виявлення замаскованих вогневих і

бронетанкових засобів радіаційними приладами з літальних апаратів

НУБІП України

Михайло Дівізінюк, Юрій Луценко, Олег Мирошник, Олег Бас,
Олександр Авраменко. *Social development and Security*, Vol. 10, No. 4,
2020

8. Multistage Analysis of Abnormal Human Behavior in Complex Scenes/ Lei

Cai ,Peien Luo, Guangfu Zhou. *Hindawi Journal of Sensors Volume 2019*,
Article ID 1276438, 10 pages.

9. Human behavioral pattern analysis-based anomaly detection system in
residential space/ Seunghyun Choi, Changgyun Kim¹, Yong-Shin Kang,

Sekyoung Youm. *The Journal of Supercomputing (2021) 77:9248–9265.*

10. Development of a methodology to predict and monitor emergency situations
of the elderly based on object detection/ Sekyoung Youm, Changgyun Kim,
Seunghyun Choi, Yong-Shin Kang. *Springer Science+Business Media, LLC*,
part of Springer Nature 2018

НУБІП України

НУБІП України

НУБІП України

НУБІП України