

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
Навчально-науковий інститут прикладного системного аналізу  
Кафедра системного проектування**

«На правах рукопису»  
УДК \_\_\_\_\_

До захисту допущено:  
Завідувач кафедри  
\_\_\_\_\_ Вадим МУХІН  
«\_\_» \_\_\_\_\_ 2021 р.


**Магістерська дисертація**

**на здобуття ступеня магістра  
за освітньо-професійною програмою**

**“Інтелектуальні сервіс-орієнтовані розподілені обчислювання”  
зі спеціальності 122 "Комп'ютерні науки"**

**на тему: « Інтеграція потоків даних BSN в єдину хмарну платформу ЕСОЗ  
(e-health)»**

Виконав :  
студент II курсу, групи ДА-01мп  
Саядов Ільмар Ісмаїл огли \_\_\_\_\_

Науковий керівник:  
Асистент, к.т.н,  
Письменний І. О. \_\_\_\_\_  


Рецензент:  
Доцент кафедри математичних методів системного аналізу, д.т.н  
Недашківська Н. І. \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних посилань.  
Студент \_\_\_\_\_,

Київ – 2021

**Національний технічний університет України  
«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Інститут/факультет Інститут прикладного системного аналізу \_\_\_\_\_  
(повна назва)

Кафедра Системного проектування \_\_\_\_\_  
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) 122 - Комп'ютерні науки \_\_\_\_\_  
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ В.Є.Мухін. \_\_\_\_\_  
(підпис) (ініціали, прізвище)

«\_\_\_» \_\_\_\_\_ 2021\_\_ р.

**ЗАВДАННЯ  
на магістерську дисертацію студенту  
(практика)**

Саядов Ільмар Ісмаїл огли \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема дисертації Інтеграція потоків даних BSN в єдину хмарну платформу ECO3 (e-health) \_\_\_\_\_

науковий керівник дисертації Письменний Ігор Олександрович, Кандидат технічних наук кафедри системного проектування ННК «ІПСА» КПІ ім. Ігоря Сікорського, Україна, Київ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «02» \_\_\_\_\_ 11 \_\_\_\_\_ 2021 р. № 3651-с

2. Строк подання студентом дисертації 20 грудня 2021 \_\_\_\_\_

3. Об'єкт дослідження

Об'єктом дослідження є Інтеграція потоків даних Body Sensor Network з використанням Дельта Лейк в хмарна платформу Електронна система охорони здоров'я eHealth

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою) \_\_\_\_\_

Предмет дослідження є розробка робочої версії інфраструктура Інтеграція потоків даних Body Sensor Network в єдину хмарну платформу ЕСОЗ (e-health) також доповнення в стару версію продукту Body Sensor Network

---

---

5. Перелік завдань, які потрібно розробити \_\_\_\_\_

1. Проаналізувати та обрати технології Big Data для вирішення проблем
  2. Розробка архітектури збору даних та зв'язків між компонентами
  3. Розробка програми інтеграції потоків даних BSN в хмарну платформу ЕСОЗ (e-health)
  4. Тестування всієї системи на вхідних даних
  5. Обробка отриманих результатів та оформлення звіту \_\_\_\_\_
- 

6. Перелік графічного (ілюстративного) матеріалу \_\_\_\_\_

Блок-схеми архітектури проекту, діаграми збору даних \_\_\_\_\_

7. Орієнтовний перелік публікацій \_\_\_\_\_

Вікіпедії — вільної енциклопедії [Електронний ресурс] Електронна система охорони здоров'я eHealth – Режим доступу: <https://uk.wikipedia.org/wiki/EHealth>.

Wireless Body Sensor Networks: A Review [Електронний ресурс] – Режим доступу:

[https://www.researchgate.net/publication/283739667\\_Wireless\\_Body\\_Sensor\\_Networks\\_A\\_Review](https://www.researchgate.net/publication/283739667_Wireless_Body_Sensor_Networks_A_Review).

Delta Lake [Електронний ресурс] – Режим доступу:

<https://databricks.com/product/delta-lake-on-databricks>. \_\_\_\_\_

---

8. Консультанти розділів дисертації\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 29.06.2021

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1.	Отримання завдання на магістерську дисертацію	01.09.2021 – 06.09.2021	
2.	Огляд технічної літератури за темою	07.09.2021 – 13.09.2021	
3.	Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі,	14.09.2021 – 20.09.2021	

---

\* Консультантом не може бути зазначено наукового керівника

	завдань, новизни, практичної значущості результатів. Дослідження актуальності обраної теми		
4.	Перший розділ. Аналіз методів та моделі інтеграції потоків даних і моделювання в режимі реального часу. Архітектура Delta Lake баз даних для запису неструктурованої інформації в архітектурі додатку .	21.09.2021 – 27.09.2021	
5.	Другий розділ. Принципи роботи системи eHealth та дослідити взаємодії з API хмарної платформи.	28.09.2021 – 04.10.2021	
7.	Третій розділ. Розробка хмарної системи VAN для вдосконалення роботи інтеграції потоків даних в хмарну платформу eHealth	05.10.2021 – 18.10.2021	
8.	Аналіз результатів	19.10.2021 – 25.10.2021	
9.	Проведення аналізу ринкових можливостей стартап – проекту	26.10.2021 – 01.11.2021	
10.	Підготовка ілюстративного матеріалу	02.11.2021 – 15.11.2021	

Студент

\_\_\_\_\_

(підпис)

Саядов І.І.

(ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_

(підпис)

Письменний І. О.

(ініціали, прізвище)

## АНОТАЦІЯ

### **Саядов І.І. Інтеграція потоків даних BSN в єдину хмарну платформу ЕСОЗ (e-health)**

Магістерська дисертація на здобуття ступеня магістра за освітньо-професійною програмою Інтелектуальні сервіс-орієнтовані розподілені обчислювання.

Даний проект був присвячений розробці додатку Body Sensor Network за допомогою хмарних обчислювань та інтеграції даних в хмарну платформу ЕСОЗ з використанням можливостей платформи Amazon Web Services.

У дисертації проведений короткий огляд предметної області, використання технологій натільних комп'ютерних мереж для допомоги в діагностиці, визначення стану пацієнта та допомоги робот медичних установ на прикладі e-health. Була створена тестова модель для реалізації поставленої задачі.

Технології на основі VAN мають великий потенціал для безперервного моніторингу в амбулаторних умовах, раннього виявлення аномальних станів і контрольованої реабілітації. Вони можуть підвищити впевненість пацієнтів і поліпшити якість життя, а також сприяти здоровій поведінці та обізнаності про здоров'я. Безперервний моніторинг з раннім виявленням, ймовірно, може забезпечити пацієнтам підвищений рівень впевненості, що, в свою чергу, може поліпшити якість життя. Крім того, амбулаторний моніторинг дозволить пацієнтам займатися звичайною повсякденною діяльністю, а не залишатися вдома або поруч зі спеціалізованими медичними службами. І останнє, але не менш важливе, включення даних безперервного моніторингу в медичні бази даних дозволить проводити комплексний аналіз всіх даних для оптимізації індивідуального догляду та забезпечення пошуку знань за допомогою інтегрованого інтелектуального аналізу даних. Дійсно, за нинішньої технологічної тенденції до інтеграції процесорів і бездротових інтерфейсів скоро з'являться інтелектуальні датчики розміром з монету. Вони будуть наноситися у

вигляді шкірних пластирів, легко інтегруватися в персональну систему моніторингу і носитися протягом тривалого періоду часу.

Ключові слова: великі дані, E-health, хмарні обчислення, натільний комп'ютерна мережа.

## **SUMMARY**

Saiadov I. I. Integration of BSN data flows into a single ECOS Cloud Platform (e-health)

Master's thesis for a master's degree in the educational and Professional Program intelligent service-oriented distributed computing.

This project was dedicated to developing The Body Sensor Network Application using cloud computing and the Amazon Web Service and integrating data into cloud platform.

The dissertation provides a brief overview of the subject area, the use of technologies of wearable computer networks to help in diagnostics, determine the patient's condition and help the robot of medical institutions on the example of e-health. A test model was created to implement this task.

BAN-based technologies have great potential for continuous monitoring in outpatient settings, early detection of abnormal conditions, and controlled rehabilitation. Continuous monitoring with early detection is likely to provide patients with increased levels of confidence, which in turn can improve their quality of life. In addition, outpatient monitoring will allow patients to engage in normal daily activities, rather than staying at home or near specialized medical services. Last but not least, the inclusion of continuous monitoring data in medical databases will allow for a comprehensive analysis of all data to optimize individual care and ensure knowledge acquisition through integrated data mining. Indeed, with the current technological trend towards integrating processors and wireless interfaces, we will soon have coin-

sized smart sensors. They will be applied as skin patches, easily integrated into a personal monitoring system, and worn for a long period of time.

Keywords: big data, E-health, cloud computing, body sensor network.

## Зміст

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ</b> .....	10
<b>ВСТУП</b> .....	11
<b>1 МЕТА ТА ОПИС ЗАВДАННЯ</b> .....	13
<b>1.1 Завдання</b> .....	13
<b>1.2 Мета магістерської дисертації</b> .....	13
<b>1.3.1 Big Data</b> .....	15
<b>1.3.2 E-Health</b> .....	16
<b>1.3.3 Обробка даних в режимі реального часу</b> .....	18
<b>1.3.4 Натільна комп'ютерна мережа</b> .....	19
<b>1.4 Висновок</b> .....	20
<b>2 АНАЛІЗ МЕТОДІВ ТА МОДЕЛІ ІНТЕГРАЦІЇ ПОТОКІВ ДАНИХ І ЇХ АНАЛІЗУ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ ТА АРХІТЕКТУРА DELTA LAKE БАЗ ДАНИХ</b> .....	21
<b>2.1 Методи обробки та аналізу даних в реальному часі</b> .....	21
<b>2.1.1 Рекомендаційні машини</b> .....	23
<b>2.1.2 Підхід на основі змісту</b> .....	24
<b>2.1.3 Спільне фільтрування</b> .....	25
<b>2.2 Методи та моделі інтеграції потоків даних</b> .....	25
<b>2.3 Delta Lake</b> .....	29
<b>3 ПРИНЦИПИ РОБОТИ СИСТЕМИ EHEALTH ТА ДОСЛІДИТИ ВЗАЄМОДІЇ З API ХМАРНОЇ ПЛАТФОРМИ</b> .....	38
<b>3.1 Структура E-Health</b> .....	38
<b>3.1.1 Електронна медична карта</b> .....	40
<b>3.1.2 Телемедицина</b> .....	41
<b>3.1.3 Абстракції E-Health</b> .....	42
<b>3.1.4 Сумісність і працездатність</b> .....	44
<b>3.1.5 Особливості пристрою і системи</b> .....	44
<b>3.1.6 Безпека в електронній охороні здоров'я</b> .....	45
<b>3.1.7 Конфіденційність</b> .....	45

<b>3.1.8</b>	<b>Абстрактна Модель Обслуговування Користувачів.....</b>	<b>46</b>
<b>3.1.9</b>	<b>Машинне навчання в E-Health.....</b>	<b>48</b>
<b>3.1.10</b>	<b>Особливості M-Health .....</b>	<b>49</b>
<b>3.2</b>	<b>Висновок .....</b>	<b>53</b>
<b>4</b>	<b>РЕАЛІЗАЦІЯ ДОДАТКА VAN З ІНТЕГРАЦІЄЮ ПОТОКІВ ДАНИХ В EHEALTH.....</b>	<b>54</b>
<b>4.1</b>	<b>Збір інформації за допомогою Amazon Kinesis.....</b>	<b>55</b>
<b>4.2</b>	<b>Обробка та аналіз даних використовуючи Delta Lake у EC2 .....</b>	<b>60</b>
<b>4.3</b>	<b>Збереження Delta Lake даних на S3 .....</b>	<b>69</b>
<b>4.4</b>	<b>Збереження даних у Amazon DynamoDB .....</b>	<b>70</b>
<b>4.4</b>	<b>Висновки .....</b>	<b>72</b>
<b>5</b>	<b>РОЗРОБКА СТАРТАП ПРОЕКТУ .....</b>	<b>73</b>
	<b>ВИСНОВКИ .....</b>	<b>97</b>
	<b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>	<b>99</b>

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

BSN – Body Sensor Network

WSN – Wireless Sensor Networks

CABAN – Cloud-assisted body area networks

DSMS – Data stream management systems

OSWA – Open Sensor Web Architecture

AWS – Amazon Web Services

EC2 - Amazon Elastic Compute Cloud

## ВСТУП

За останні два десятиліття інформаційно-комунікаційні технології суттєво змінили розвиток відносин у всіх сферах суспільного життя. Сучасна інформаційна епоха надала необмежені можливості для вільного обміну інформацією, що мають суспільне значення. Інформаційно-комунікаційні технології (ІКТ) все частіше використовуються в охороні здоров'я. Одним з основних факторів цієї тенденції є важливість інформації в медичній практиці. Медичні працівники отримують та використовують інформацію про стан здоров'я пацієнтів, це має значний вплив на якість медичної допомоги. Сама інформація, її відбір і повнота стають вирішальним фактором поведінки учасників національних систем охорони здоров'я-державних органів, медичних установ, професійних компаній, організацій пацієнтів та інших. У цьому сенсі інтеграція інформаційних систем для електронного обміну даними надає величезні можливості для швидкого та ефективного управління медичною інформацією і, отже, для більш ефективної роботи медичних установ і медичних працівників. Саме тому одним з найважливіших інструментів підвищення ефективності будь-якої національної системи охорони здоров'я є побудова електронної охорони здоров'я (e-health). Електронна охорона здоров'я є цінним інструментом для вдосконалення національних систем охорони здоров'я та надання доступних і високоякісних медичних послуг. Успішне створення та використання електронної охорони здоров'я покращує комунікаційне середовище та забезпечує ряд переваг для відповідної національної системи охорони здоров'я, а також для всієї спільноти. На додаток до створення електронної охорони здоров'я, ефективна система охорони здоров'я також вимагає створення електронної медичної картки для кожного пацієнта, також називається електронною медичною картою. Електронна медична карта-це запис, що містить певну медичну інформацію для однієї людини у форматі, придатному для комп'ютерної обробки та використання. Створення електронних медичних карт приносить велику користь як пацієнтам, так і медичним працівникам. Наприклад, впровадження електронних медичних карт забезпечує швидкий і легкий доступ лікарів до медичних записів пацієнтів, дозволяючи їм

відстежувати всі важливі обставини і зміни в стані здоров'я пацієнта, а також адаптувати їх для проведення конкретних діагностичних і лікувальних заходів. Серед наявних технологій чільне місце займають технології натільної комп'ютерної мережі.

Натільна комп'ютерна мережа (BAN) с кожним роком поширюється у різні сфери діяльності людини. Ця технологія застосовується для багатьох областей, орієнтованих на людину, таких як охорона здоров'я, спорт, фітнес, військова сфера, оздоровчий стан, ергономіка, надзвичайні ситуації, безпека, безпека та соціальність. Система BAN, яка в основному складається з бездротових носимих вузлів датчиків, зазвичай координованих статичним або мобільним пристроєм, в основному експлуатується для моніторингу одиночних допоміжних ліній. Дані, генеровані BAN, можуть бути оброблені в режимі реального часу координатором BAN або передані на сторону сервера для он-лайн, офлайн-обробки та тривалого зберігання. Мережа BAN, яку носить громада людей, виробляє велику кількість контекстних даних, які потребують масштабованого та ефективного підходу для розробки та зберігання. Хмарні обчислення можуть забезпечити гнучку інфраструктуру зберігання та обробки для виконання як онлайн, так і офлайн аналізу потоків даних датчиків тіла. Запровадження BAN з використанням хмари супроводжується разом із основними проблемами, які необхідно вирішити для їх розвитку та управління. Поточний стан сучасних технологій переглядається та оформляється відповідно до основних вимог до ефективних архітектурних технологій BAN. Нарешті, обговорюються відповідні відкриті дослідницькі питання з точки зору ефективності, масштабованості, безпеки, сумісності, прототипування, динамічного розгортання та управління.

# 1 МЕТА ТА ОПИС ЗАВДАННЯ

## 1.1 Завдання

Магістерська дисертація має наступні завдання:

1. Проаналізувати та обрати технології Big Data для вирішення проблем
2. Розробка архітектури збору даних та зв'язків між компонентами.
3. Розробка програми інтеграції потоків даних BSN в хмарну платформу ECO3 (e-health)
4. Тестування всієї системи на вхідних даних
5. Обробка отриманих результатів та оформлення звіту

## 1.2 Мета магістерської дисертації

Мета роботи полягає у проектуванні інфраструктури обробки даних Body Sensor Network з інтеграцією потоків в систему e-health.

**Великі дані** (англ. Big Data) – це галузь, яка розглядає способи аналізу, систематичного отримання інформації або іншим чином отримання з набору даних, які є занадто великими або складними, щоб їх вирішувати, використовують традиційне програмне забезпечення для обробки даних. Дані з багатьма випадками (рядки) пропонують більш статистичну потужність, тоді як дані з більшою складністю (більше атрибутів або стовпців) можуть призвести до більш високого показника помилкового виявлення. Проблеми великих даних включають збір даних, зберігання даних, аналіз даних, пошук, обмін, передачу, візуалізація, запити, оновлення, конфіденційність інформації та джерело даних. Великі дані спочатку були пов'язані з трьома ключовими поняттями: об'єм, різноманітність та швидкість. Коли ми обробляємо великі дані, ми можемо не робити вибірку, а просто спостерігати та відстежувати, що відбувається. Тому великі дані часто включають дані з розмірами, що перевищують можливості традиційного програмного забезпечення для обробки протягом прийняттого часу та значення.[3].

**Електронна охорона здоров'я** - це відносно недавня практика охорони здоров'я, забезпечена електронними процесами і комунікаціями, починаючи, принаймні, з 1999

року. Використання терміна варіюється, оскільки він охоплює не тільки "інтернет-медицину", як це було задумано в той час, але також "практично все, що пов'язано з комп'ютерами та медициною". Дослідження, проведене в 2005 році, виявило 51 унікальне визначення. Деякі стверджують, що він взаємозамінний з інформатикою охорони здоров'я з широким визначенням, що охоплює електронні/цифрові процеси в охороні здоров'я, в той час як інші використовують його в більш вузькому сенсі медичної практики з використанням Інтернету. Він також може включати додатки для охорони здоров'я та посилання на мобільні телефони, звані mHealth або m-Health.

**Натільна комп'ютерна мережа (BSN)** також називається бездротовою мережею зони тіла (WBAN) або мережею медичної зони тіла (MBAN), являє собою бездротову мережу носяться обчислювальних пристроїв. Пристрої можуть бути вбудовані в тіло у вигляді імплантатів, можуть бути встановлені на поверхні тіла у фіксованому положенні або можуть супроводжуватися пристроями, які люди можуть носити в різних положеннях. У той час як спостерігається тенденція до мініатюризації пристроїв, зокрема, мережі області тіла складаються з декількох мініатюрних датчиків тіла (BSU) разом з одним центральним блоком тіла (BCU). Інтелектуальні пристрої розміром дециметра (вкладка та панель) все ще відіграють важливу роль з точки зору виконання функцій концентратора даних або шлюзу даних та надання інтерфейсу користувача для перегляду та керування додатками BAN на місці. Розробка технології WBAN почалася приблизно в 1995 році з ідеї використання технологій бездротової персональної мережі (WPAN) для здійснення зв'язку на тілі людини, поблизу нього і навколо нього. Приблизно через шість років термін став ставитися до систем, в яких зв'язок повністю здійснюється всередині, на і в безпосередній близькості від людського тіла. Система WBAN може використовувати бездротові технології WPAN як шлюзи для досягнення великих відстаней. За допомогою шлюзових пристроїв можна підключати носяться пристрої на тілі людини до Інтернету. Таким чином, медичні працівники можуть отримати доступ до даних пацієнта онлайн за допомогою Інтернету незалежно від місця розташування пацієнта.

**Amazon Web Services** (AWS) – це сервіс компанії Amazon, яка надає свої технології для хмарних обчислень на замовлення фізичним особам, компаніям та урядам. У сукупності ці веб-сервіси хмарних обчислень забезпечують набір примітивної абстрактної технічної інфраструктури та розподілених будівельних блоків та інструментів. Один з таких сервісів - Amazon Elastic Compute Cloud (EC2), який дозволяє користувачам мати в своєму розпорядженні віртуальну кластерну мережу комп'ютерів, доступну весь час через Інтернет. Версія віртуальних комп'ютерів AWS емулює більшість атрибутів реального комп'ютера, включаючи апаратні центральні процесорні блоки (процесори) та графічні одиниці обробки (GPU) для обробки, оперативну пам'ять, накопичувач на жорсткому диску, вибір операційних систем, мережу, попередньо завантажене прикладне програмне забезпечення, такі як веб-сервери, бази даних та управління відносинами з клієнтами (CRM).[21]

## **1.3 Дослідження предметної області**

### **1.3.1 Big Data**

Великі дані відносяться до масивних складних структурованих і неструктурованих наборів даних, які швидко генеруються і передаються з самих різних джерел. Ці атрибути складають три основні аспекти великих даних:

**Обсяг:** величезні обсяги збережених даних.

**Швидкість:** блискавична швидкість, з якою потоки даних повинні оброблятися і аналізуватися.

**Різноманітність:** різні джерела та форми, з яких збираються дані, такі як цифри, текст, відео, зображення, аудіо та текст.

В наші дні дані постійно генеруються кожного разу, коли ми відкриваємо додаток, шукаємо в Google або просто подорожуємо з місця на місце за допомогою наших мобільних пристроїв. Масивні колекції цінної інформації, необхідної компаніям і організаціям для управління, зберігання, візуалізації та аналізу.

Традиційні інструменти обробки даних не пристосовані для обробки такого роду складності і обсягу, що призвело до появи безлічі спеціалізованих програмних засобів для обробки великих даних і архітектурних рішень, призначених для управління навантаженням.

Різноманітність великих даних робить їх за своєю суттю складними, що призводить до необхідності в системах, здатних обробляти їх різні структурні та семантичні відмінності.

Для великих даних потрібні спеціалізовані бази даних NoSQL, які можуть зберігати дані таким чином, щоб не було потрібно суворого дотримання певної моделі. Це забезпечує гнучкість, необхідну для послідовного аналізу, здавалося б, розрізнених джерел інформації, щоб отримати цілісне уявлення про те, що відбувається, як діяти і коли діяти.

При агрегуванні, обробці та аналізі великих обсягів даних їх часто класифікують як оперативні або аналітичні дані і зберігають відповідним чином. Операційні системи обслуговують великі пакети даних на декількох серверах і включають такі вхідні дані, як інвентаризація, дані про клієнтів і покупках — повсякденна інформація всередині організації.

### **1.3.2 E-Health**

Портативні пристрої, такі як смартфони і мобільний доступ в інтернет, набули поширення в останні десятиліття. Термін "Електронна охорона здоров'я" означає електронну охорону здоров'я. Інструменти, включені в концепцію електронної охорони здоров'я, використовують телефони, комп'ютери, Інтернет та пов'язані з ними програми для поліпшення галузі охорони здоров'я. Впровадження технологій електронного охорони здоров'я було задокументовано для лікування різних хронічних захворювань, включаючи астму та алергічні стани. Пацієнти отримали можливість спілкуватися новими способами, які можна було б використовувати економічно ефективно для поліпшення контролю захворювань і якості життя постраждалих. Крім того, ці інновації відкривають нові можливості для академічних

дослідників. Наприклад, Електронна охорона здоров'я дозволила дослідникам збирати дані, які раніше були недоступні або важкодоступні, і аналізувати їх за допомогою нових інструментів, в сукупності званих "великими даними". Роль електронної охорони здоров'я стала більш важливою з початку 2020 року через правила фізичного дистанціювання та обмеження на мобільність, які були застосовані у всьому світі у відповідь на пандемію коронавірусної хвороби 2019 року.

Використання великих даних в охороні здоров'я процвітає. Нещодавно було випущено кілька нових інновацій та методів, які застосовують великі дані для біоінформатики та досліджень у галузі охорони здоров'я. В даний час такі пристрої, як смартфони, датчики; окремі особи, такі як пацієнти, медичні працівники або дослідники, а також колективні організації, такі як медичні установи та установи, регулярно створюють і збирають величезну кількість медичних даних, які дозволяють діагностувати і лікувати нові розлади. Справжня дилема в застосуванні мобільного охорони здоров'я полягає в тому, як виявляти, збирати, аналізувати і поширювати інформацію, щоб поліпшити і спростити життя людей за допомогою раннього прогнозування факторів ризику. Наприклад, дослідники розробили безліч методів, що дозволяють управляти хронічними захворюваннями. Медичні пристрої для постійного моніторингу показників здоров'я або своєчасного виявлення онлайн-медичних даних, таких як самостійна фізіотерапія, вкрай необхідні. Сьогодні для доступу до високошвидкісного Інтернету на мобільних телефонах кілька розумних пацієнтів використовують додатки для смартфонів (додатки) для регулярного моніторингу різних щоденних потреб у галузі охорони здоров'я. Такі програми та пристрої для смартфонів все частіше використовуються та впроваджуються в охорону здоров'я та телемедицину через Інтернет речей (IoT). Однак конфіденційність і безпека є основними проблемами, які вимагають співпраці з директивними органами і оперативного інформування про потенційні небезпеки. У цій роботі розглядаються програми Інтернету речей і m-health в охороні здоров'я. Вона також описує інноваційні методи поліпшення здоров'я за допомогою

обчислювальних технологій та методів інтеграції Інтернету речей та m-health, включаючи запропоновану модель для самостійного управління діабетом.

### **1.3.3 Обробка даних в режимі реального часу**

Обробка в режимі реального часу виконується для потоків даних, одержуваних в реальному часі і оброблюваних з мінімальною затримкою для створення звітів або автоматизованого реагування в режимі реального часу (або наближеному до реального часу). Наприклад, рішення для моніторингу дорожнього руху в режимі реального часу дозволяє відстежувати зони інтенсивного руху за даними від датчиків. На основі цих даних можна динамічно оновлювати карту з інформацією про пробки або автоматично регулювати рух за допомогою спеціальних смуг або інших систем управління дорожнім рухом. Обробкою в режимі реального часу вважається обробка неприв'язаного потоку вхідних даних з мінімальним часом затримки — кілька мілісекунд або секунд. Ці вхідні дані зазвичай надходять у неструктурованому або напівструктурованому форматі, наприклад JSON. Вимоги до обробки тут такі ж, як і в разі пакетної обробки, крім додаткового обмеження за часом для підтримки використання в режимі реального часу. Оброблені дані зазвичай зберігаються в сховищі аналітичних даних, яке оптимізовано для аналізу та візуалізації. Також оброблені дані можна безпосередньо приймати на шарах аналітичної обробки та звітності для аналізу, бізнес-аналітики та візуалізації на панелях моніторингу в режимі реального часу.

Однією з головних проблем в будь-якій системі обробки в режимі реального часу є підтримка можливості прийому і збереження повідомлень в режимі реального часу, особливо при великих обсягах даних. Обробка повинна виконуватися так, щоб не блокувати конвеєр прийому даних. Потрібно використовувати сховище даних, що підтримує операції запису у великих обсягах. Ще одна проблема-створення можливостей для швидкого реагування, наприклад для створення оповіщень в режимі реального часу або представлення даних на панелі моніторингу в режимі реального часу (або наближеному до реального часу).

### 1.3.4 Натільна комп'ютерна мережа

Тілесна мережа (BAN), також називається бездротовою тілесною мережею (WBAN) або мережею датчиків тіла (BSN) або медичною тілесною мережею (MBAN), являє собою бездротову мережу носяться обчислювальних пристроїв. пристрої заборони можуть бути вбудовані в тіло у вигляді імплантатів, можуть бути встановлені на поверхні тіла у фіксованому положенні або можуть супроводжуватися пристроями, які люди можуть носити в різних положеннях, наприклад, в кишенях одягу, вручну або в різних сумках.[6]. Хоча спостерігається тенденція до мініатюризації пристроїв, зокрема, тілесні мережі складаються з декількох мініатюрних блоків датчиків тіла (BSU) разом з єдиним центральним блоком тіла (BCU)[7]. інтелектуальні пристрої більшого дециметрового розміру (вкладка і панель) як і раніше відіграють важливу роль в якості концентратора даних або шлюзу даних і надають користувальницький інтерфейс для перегляду і управління додатками BAN на місці. Розробка технології WBAN почалася приблизно в 1995 році з ідеї використання технологій бездротової персональної мережі (WPAN) для здійснення зв'язку на тілі людини, поблизу нього і навколо нього. Приблизно через шість років термін "заборона" став ставитися до систем, в яких зв'язок повністю здійснюється всередині, на і в безпосередній близькості від людського тіла. система WBAN може використовувати бездротові технології WPAN як шлюзи для досягнення великих відстаней. За допомогою шлюзових пристроїв можна підключати носяться пристрої на тілі людини до Інтернету. Таким чином, медичні працівники можуть отримати доступ до даних пацієнта онлайн за допомогою Інтернету незалежно від місця розташування пацієнта.[1]

Очікується, що початкові застосування заборон з'являться в першу чергу в сфері охорони здоров'я, особливо для безперервного моніторингу та реєстрації життєво важливих параметрів пацієнтів, які страждають хронічними захворюваннями, такими як діабет, астма і серцеві напади.

Введений щодо пацієнта заборона може попередити лікарню ще до того, як у нього трапиться Серцевий Напад, шляхом вимірювання змін його життєво важливих

показників. Заборона пацієнтам з діабетом може автоматично вводити інсулін за допомогою помпи, як тільки рівень їх інсуліну знизиться. Заборона може бути використана для вивчення основних змін стану здоров'я та динаміки захворювання[5]. Інші сфери застосування цієї технології включають спорт, військові або безпеку. Поширення технології на нові області також може сприяти комунікації шляхом безперешкодного обміну інформацією між окремими особами або між окремими особами і машинами.

Останнім міжнародним стандартом для заборон є стандарт IEEE 802.15.6.[11]

Типова заборона або BSN вимагає датчиків моніторингу життєво важливих показників, детекторів руху( за допомогою акселерометрів), щоб допомогти визначити місце розташування контрольованої людини і деяку форму зв'язку, для передачі показань життєво важливих показників і руху практикуючим лікарям або особам, які надають медичну допомогу. Типовий комплект локальної мережі буде складатися з датчиків, процесора, приймача і акумулятора. Були розроблені фізіологічні датчики, такі як датчики ЕКГ і SpO<sub>2</sub>. Інші датчики, такі як датчик артеріального тиску, датчик ЕЕГ і КПК для інтерфейсу BSN, знаходяться в стадії розробки.[12]

## **1.4 Висновок**

В даному розділі була сформульована мета та завдання магістерської дисертації, коротко описані концепції, походження, зони застосування та інтеграція з eHealth системи BSN. Особливо ми зосереджуємось на статусі досліджень, тенденціях розвитку та викликах сенсорної технології, технології синтезу даних та інтеграції цих даних у інші системи. Також була досліджена предметна область проекту Body Sensor Network, розглянута система E-Health та обробка і аналіз даних в режимі реального часу. Ще було досліджено моделі обслуговування та існуючі рішення хмарних обчислень. Було розглянуто детально мету натільної комп'ютерної мережі та основні потреби інтеграції для хорошої роботи усієї інфраструктури даних.

## **2 АНАЛІЗ МЕТОДІВ ТА МОДЕЛІ ІНТЕГРАЦІЇ ПОТОКІВ ДАНИХ І ЇХ АНАЛІЗУ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ ТА АРХІТЕКТУРА DELTA LAKE БАЗ ДАНИХ.**

У цьому розділі розглянемо методи обробки даних в режимі реального часу та методів інтеграції інформаційних систем і потоків даних. Розглянемо різні методи збору та обробки даних в режимі реального часу, розглянемо методи та випадки інтеграції потоків даних. Також розглянемо архітектуру Delta Lake бази даних для збереження неструктурованої інформації, також розглянемо сховища які надає нам дана технологія .

### **2.1 Методи обробки та аналізу даних в реальному часі**

Більшість методів обробки та аналізу даних є статичними. Це означає, що з їх допомогою виконується аналіз історичних даних (тобто даних, накопичених раніше і незмінних в процесі аналізу), як для опису властивостей даних, так і для побудови моделей передбачення на основі навчальної вибірки. Аналіз даних в реальному часі (або Real-Time Analytics), в основному, відносяться до задачі передбачення. На відміну від статичних методів вони навчаються динамічно і засновані на зворотному зв'язку від прогнозу, отриманого за допомогою передбачувальної моделі (постійному перенавчанні). Можливий шлях досягти цього-зібрати всі дані (постійно додавати до даних, що піддаються аналізу, нові дані) і застосовувати алгоритми Data Mining, починаючи з початкового (можливо незначного) набору даних і закінчуючи наявними в поточний момент часу повним набором даних. Така процедура називається накопичуване навчання (Batch Learning). Однак накопичуване навчання часто не ефективно, тому що. вимагає запам'ятовування великих обсягів даних і призводить до тривалого періоду навчання. З цієї причини найчастіше додатки реального часу для оновлення існуючих моделей використовують тільки нові дані.[27]

Методи обробки даних в реальному часі стали надзвичайно популярні в останні роки і, безсумнівно, займають лідируюче положення. Однак сам по собі адаптивний

підхід не обмежується тільки областю Big Data, він знаходить більш широке застосування в багатьох наукових областях.

Прикладом використання адаптивного підходу є рішення диференціальних рівнянь, особливо приватних диференціальних рівнянь-однієї з класичних областей математики. [27]

Через їх складність, більшість диференціальних рівнянь не може бути вирішено аналітично, тому часто для їх вирішення використовують методи апроксимації. Найпопулярніші схеми апроксимації-метод кінцевого елемента (Finite Element Method — FEM) і метод кінцевих різниць (Finite Difference Method — FDM). Методи кінцевого елемента апроксимують вирішенням функцій  $F$  диференціальних рівнянь через базиси на решітках. Чим щільніше решітка, тим краще апроксимація; з іншого боку, більш щільні решітки вимагають більш високого рівня обчислень [28].

Конвергенція і збіжність швидкості апроксимації безпосередньо залежать від регулярності проблеми; особливо від геометричної форми області (умова Ліфшиця, уніфікована конічна умова тощо), так само, як і від специфічних прикордонних умов.

У більшості додатків зроблені "класичні" припущення регулярності наближені до дійсних, і ітераційні методи кінцевих елементів швидко сходяться в одну точку. Однак іноді ми маємо грубу специфічність і надзвичайно складні прикордонні умови, які важко класифікувати. Це породжує багато проблем. Чи є межі спотвореної заготовки дійсно умовою Ліфшиця? Чи є зворотний тиск дійсно функцією Дірака або скоріше простором.

В результаті, на практиці такі проблеми часто призводять до повільної конвергенції з непридатним результатом. Для того щоб подолати ці проблеми, застосовуються адаптивні методи кінцевих елементів, які використовують покроковий підхід, щоб побудувати решітки і вирішити рівняння, що з'являються в результаті застосування методів кінцевих елементів. Загальна послідовність дій включає в себе наступні кроки:

1. Генерація початкової моделі (наприклад, на основі наявних не оновлюваних даних).

2. Грунтуючись на моделі вибрати оптимізаційну дію (рекомендація, Пошта і т.п.).

3. Аналіз відповіді та обчислення помилки.

4. Якщо мета досягнута (наприклад, вся продукція продана чи немає нових даних), то завершити обчислення.

5. Оновити модель видобутку даних, грунтуючись на помилку відповіді.

6. Перехід до кроку 2.

Слід зазначити, що використання адаптивної процедури має важливу відмінність. Не тільки конструюється точна модель (як в методі кінцевого елемента), але і плануємо досягти бізнес-мети!

Щоб краще зрозуміти різницю, розглянемо приклад використання адаптивного підходу в call-центрі компанії, що просуває новий товар. Агенти компанії мають список потенційних клієнтів з різними характеристиками (стать, адреса, вік та ін.) і прагнуть звертатися тільки до тих клієнтів, які, найімовірніше, замовлять новий продукт безпосередньо по телефону. Це типова задача класифікації (див.гл. 5). Застосування адаптивного підходу означало б, що після кожного дзвінка (або ряду дзвінків) використовуються відповіді клієнтів для оновлення моделі класифікації, яка потім використовується повторно для нового передбачення потенційних клієнтів.

В результаті використання адаптивного підходу буде побудована модель класифікації, яка досить точно визначає клієнта на підставі його характеристик. Слід зазначити, однак, тут істотне протиріччя з бізнес-метою: обдзвонювати тільки тих клієнтів, які дійсно зроблять замовлення (не витрачаючи гроші на дзвінки людям, які замовлення не зроблять). З іншого боку, для того щоб побудувати частину класифікаційної моделі з низькою ймовірністю замовлення, необхідно також обдзвонювати людей, які не роблять замовлення, що і суперечить бізнес-цілі.

### **2.1.1 Рекомендаційні машини**

В даний час існує велика кількість різних підходів до побудови рекомендаційних машин, що ускладнює їх повну характеристику. Необхідно відзначити також, що багато з рекомендаційних машин засновані на гібридних підходах.

У рекомендаційних машинах використовуються наступні способи збору даних:

- явний збір даних-система просить користувача оцінювати пункти або заповнювати опитувальні анкети;
- неявний збір даних — система спостерігає за діями користувача (кляцання, закупівлі і т.д.).

Очевидно, неявний збір даних-найважливіший.

Формування рекомендацій може відбуватися на основі наступних підходів:

- на основі змісту — рекомендації генеруються на основі вивчення змісту елементів (продуктів);
- на основі транзакції — рекомендації генеруються на основі Користувацької поведінки.

Історично перші рекомендаційні системи використовували метод спільного фільтрування який ґрунтується на транзакціях і є дуже популярним на сьогоднішній день. Через свою популярність спільне фільтрування іноді використовується як синонім для всіх рекомендаційних систем. За аналогією підхід на основі змісту іноді називають фільтруванням на основі змісту, тоді як підхід на основі транзакцій посилається на метод спільного фільтрування.

### **2.1.2 Підхід на основі змісту**

Підхід на основі змісту використовує такі властивості елементів, як назва, опис, Ціна з тим, щоб обчислити ступінь подібності між ними. При цьому зазвичай застосовують кластеризацію і методи Text Mining. В даному випадку, якщо користувач вибирає деякий елемент, йому рекомендуються елементи, подібні обраному.

Успішними прикладами використання рекомендаційних систем, побудованих на основі змісту, є системи рекомендації музики, які рекомендують пісні, засновані на аналізі звукової хвилі музики. Відомий приклад-пошук музики і рекомендації на сайті Spotify.

У загальному випадку рекомендації, побудовані на основі змісту, мають більш низьку прогнозуючу здатність, ніж рекомендації на основі транзакцій, які будуть розглянуті далі. Однак для нових елементів (які не мають ніякої великої операційної історії і з якими не пов'язані будь-які транзакції) подібні системи часто бувають дуже корисні. Найбільш корисні рекомендації, отримані на основі комбінації двох позначених раніше підходів.

### 2.1.3 Спільне фільтрування

Спільне фільтрування (Collaborative Filtering, CF) є одним з перших методів генерації рекомендацій. Першою системою, яка використовувала спільне фільтрування, була система інформаційний Гобелен (Information Tapestry). Даний проект був розроблений Хероу PARC на початку 90-х років [28].

Спільне фільтрування є методом інформаційного пошуку (IR), який використовує набір явно або неявно зібраних/отриманих користувацьких переваг, як міру інформаційної якості/доречності. У більш загальному вигляді CF близький до методу найближчого сусіда.

## 2.2 Методи та моделі інтеграції потоків даних

Розвиток інформаційної сфери спричинило за собою інформатизацію суспільства. В даний час активно відбувається автоматизація процесів в різних видах діяльності. Яскравими прикладами є такі проекти, як «Портал держпослуг», «електронний щоденник», які дозволяють виконувати різні дії, такі як оплата комунальних послуг, запис до лікаря, відстеження успішності школяра, не виходячи з дому. У зв'язку з цим необхідно розуміти, які з методів інтеграції інформаційних систем є оптимальними для кожного конкретного випадку.

Розглянемо найбільш популярні і використовувані методи інтеграції.

**Інтеграція на рівні брокерів.** Перевагою даного методу є універсальність: як правило, в будь-якій ситуації можна реалізувати додатковий програмний модуль, який може звертатися в інші системи різними способами. Наприклад, такий модуль

може звертатися до однієї системи через базу даних (БД), а до іншої за допомогою RPC (англ. Remote Procedure Call- віддалений виклик процедур). Недоліками такого підходу інтеграції є трудомісткість і складність реалізації, і, як наслідок, висока вартість розробки, впровадження та підтримки.

**Інтеграція на рівні інтерфейсів** (фізичних, програмних і призначених для користувача). Метою було об'єднання розподілених програмних додатків, реалізованих різними розробниками в різний час, в подобу єдиної системи. Додатки зв'язувалися за принципом "кожен з кожним", що, в кінцевому підсумку, ускладнювало їх взаємодію і створювало ряд проблем і помилок. Також ускладнювалося використання успадкованих (Legacy Software) і вбудованих (Embedded System) систем. Описаний підхід інтеграції зручний для невеликої кількості програмних додатків. Для великого числа додатків він є малоефективним і не забезпечує побудову якісно нових запитів до об'єднаних даних. Таким чином, агрегування даних не принесе вираш. На даний момент, проблема інтеграції на рівні інтерфейсів вирішується на базі впровадження інформаційних підсистем, які реалізуються стандартними додатками з відкритими програмними інтерфейсами (англ. Open Application Programming Interface, OAPI-відкритий програмний інтерфейс програми, відкритий інтерфейс прикладного програмування).

**Інтеграція на функціонально-прикладному та організаційному рівнях.** Даний вид інтеграції побудований на об'єднанні декількох однотипних або схожих функцій в макрофункції, в яких перерозподіляються ресурси, потоки даних, управління і механізми виконання. Як наслідок, Це тягне за собою реорганізацію інформаційних структур, бізнес-процесів і, відповідно, перебудову схем їх інформаційного та документаційного забезпечення. Переваги даного виду інтеграції:

- прозорість і керованість процесів;
- процеси стають менш витратними;
- скорочується кількість обслуговуючого персоналу
- скорочується число помилок.

Недоліком інтеграції такого виду є значна трансформація або комплексний реінжиніринг всієї мережі процесів, що може спричинити за собою певні ризики. Доцільно проводити дану інтеграцію в разі, якщо організація готується до впровадження корпоративної інформаційної системи (кіс) на платформі популярного рішення. Це, в свою чергу, вимагає уніфікації і приведення бізнес-процесів до певного стандарту. Або якщо організація перебудовує свою діяльність у зв'язку зі зміною пріоритетів, розширенням і освоєнням нових сегментів ринку.

**Інтеграція на рівні корпоративних програмних додатків.** Даний вид інтеграції передбачає спільне використання виконуваного коду, а не тільки внутрішніх даних інтегрованих додатків. Програми діляться на компоненти, які потім інтегруються за допомогою стандартизованих програмних інтерфейсів (API) і спеціалізованого сполучного програмного забезпечення (ПЗ). Такий підхід дозволяє створити з цих компонентів універсальну програмну платформу (ядро), яка може бути використано всіма додатками. Кожна програма матиме лише один інтерфейс для взаємодії з цим ядром, що значно полегшує завдання інтеграції. Систему, побудовану на такому інтеграційному підході, легше адмініструвати, підтримувати і масштабувати. Можливість повторного використання функцій в рамках наявного середовища дозволяє істотно скоротити терміни і вартість розробки додатків. Обов'язковим етапом оцінки можливості інтеграції додатків, які передбачається пов'язувати в рамках певного проекту, є аналіз внутрішньої архітектури додатків. Цей аналіз може бути ускладнений тим, що, як правило, Розробник додатків, які є готовими програмними продуктами, не розкриває всіх деталей внутрішньої структури додатків.

**Інтеграція за допомогою Web-сервісів.** Даний вид інтеграції є передовим і стрімко розвивається підходом до інтеграції додатків. Він базується на наданні стандартного для Web-служб інтерфейсу доступу до додатків і їх даних. Прикладом може бути стандартний протокол доступу до об'єктів - SOAP (англ. Simple Object Access Protocol - простий протокол доступу до об'єктів). Так, за допомогою SOAP, браузер користувача може одночасно порівняти дані на декількох обраних веб-сайтах і представити клієнту порівняльний звіт. Інший приклад: співробітники одного

географічно розподіленого підприємства можуть одноразово використовувати корпоративні додатки, доступ до яких здійснюється через відповідні Web-сервіси (портальне рішення). Web-сервіси схожі на підхід EAI, але з однією головною відмінністю — EAI-рішення, в своїй безлічі, випускаються як окремі випадки для зв'язку певних продуктів. Відповідно, підключити до вже використовуваного EAI-рішення ще одну сторонню систему буде досить важким і тривалим завданням. За своєю природою Web-сервіси істотно більш Уніфіковані і стандартизовані. Оскільки Webсервіси базуються на загальних і єдиних для Консорціуму Всесвітньої павутини (англ. World Wide Web Consortium, W3C-Консорціум) стандартах, вони можуть працювати скрізь, де використовується мережа Інтернет.

**Інтеграція на рівні даних.** Даний вид інтеграції має на увазі, що кілька програмних додатків можуть звертатися до однієї бази даних або в кілька баз даних, пов'язаних реплікаціями. Перевагою такого виду є низька вартість інтеграції. До недоліків можна віднести наступне: якщо база даних не екранована збереженими процедурами і не має необхідних обмежень і захисту цілісності (наприклад, у вигляді вказівки каскадних операцій і тригерів), то взаємодія різних додатків з даної БД може стати причиною помилок і приводити дані в суперечливі стани. У разі якщо БД екранована і підтримується цілісність збережених даних, то в одночасно взаємодіючих з однією БД додатках будуть дублюватися частини програмного коду, що виконують однакові або схожі операції. Крім того, при внесенні змін до структури бази, необхідно окремо переписувати програмний код всіх додатків, що працюють з такою БД.

**Інтеграція на рівні сервісів.** Даний вид інтеграції заснований на фіксації інтерфейсів і форматів даних з обох сторін. Перевагою є організація стрімкого відпрацювання міжкорпоративної бізнес-логіки. Такий підхід інтеграції має і недолік: оскільки спочатку він базується на «запам'ятовуванні» або фіксації, то при зміні даних, структур або процесів утворюються проблеми і помилки, що призводить до розробки вузьконаправлених, приватних рішень. - Інтеграція на рівні користувача. Даний вид відноситься до неавтоматизованої інтеграції, і заснований на взаємодії

користувачів один з одним: обмін даними і файлами між системами через ручне копіювання, оправлення пошти і т. д. Є найбільш простими видом\підходом, і часто застосовуються в той момент, коли відбувається підготовка впровадження програмних систем, а діяльність компанії не може перериватися.

## 2.3 Delta Lake

Delta Lake-це рівень сховища, який забезпечує надійність озер даних, побудованих на HDFS і хмарних сховищах, забезпечуючи транзакції ACID за рахунок оптимістичного контролю паралелізму між записами і ізоляції моментальних знімків для узгодженого читання під час запису. Delta Lake також надає вбудоване управління версіями даних для легкого відкату і відтворення звітів. [26]

Озеро Дельта пропонує наступні ключові функції:

- Транзакції ACID: Delta Lake забезпечує транзакції ACID між кількома записами. Кожен запис є транзакцією, і існує послідовний порядок записів, записаних в журналі транзакцій. Журнал транзакцій відстежує записи на рівні файлів і використовує паралелізм, який ідеально підходить для озер даних, оскільки багаторазові записи, які намагаються змінити одні й ті ж файли, відбуваються не так часто. У сценаріях, де виникає конфлікт, Delta Lake створює виняток одночасної зміни, щоб користувачі могли обробити їх і повторити свої завдання. Delta Lake також пропонує високий рівень серіалізуемой ізоляції, який дозволяє інженерам безперервно записувати дані в каталог або таблицю, а споживачам - продовжувати читання з одного і того ж каталогу або таблиці. Читачі побачать останній запис, який існував на момент початку читання.
- Управління схемою: Delta Lake автоматично перевіряє Сумісність схеми записуваного фрейма даних зі схемою таблиці. Стівпцям, які присутні в таблиці, але не в фреймі даних, присвоюється значення null. Якщо у кадрі даних є додаткові стівпці, яких немає в таблиці, Ця операція створює виняток. У Delta

Lake є DDL для явного додавання нових стовпців і можливість автоматичного оновлення схеми.

- Масштабована обробка метаданих: Delta Lake зберігає інформацію про метадані таблиці або каталогу в журналі транзакцій замість метастору. Це дозволяє Delta Lake перераховувати файли у великих каталогах в постійний час і ефективно зчитувати дані.
- Управління версіями даних і переміщення в часі: Delta Lake дозволяє користувачам зчитувати попередній знімок таблиці або каталогу. Коли файли змінюються під час запису, Delta Lake створює новіші версії файлів і зберігає старі версії. Коли користувачі хочуть прочитати старі версії таблиці або каталогу, вони можуть надати тимчасову мітку або номер версії API для читання Apache Spark, і Delta Lake створює повний знімок станом на цю тимчасову мітку або версію на основі інформації в журналі транзакцій. Це дозволяє користувачам відтворювати експерименти і звіти, а також при необхідності повертати таблицю до її більш старих версій.
- Уніфікований приймач пакетної і потокової передачі: крім пакетної записи, Delta Lake також може використовуватися в якості ефективного приймача потокової передачі за допомогою структурованої потокової передачі Apache Spark. У поєднанні з транзакціями ACID та масштабованою обробкою метаданих, ефективний приймач потокового передавання тепер дозволяє використовувати багато варіантів аналітики практично в режимі реального часу без необхідності підтримувати складний конвеєр потокового передавання та пакетної обробки.
- Оновлення та видалення записів (найближчим часом): Delta Lake буде підтримувати команди злиття, оновлення та видалення DML. Це дозволяє інженерам легко оновлювати і видаляти записи в озерах даних і спрощувати збір даних про зміни і варіанти використання GDPR. Оскільки Delta Lake відстежує і змінює дані з деталізацією на рівні файлів, це набагато ефективніше, ніж читання і перезапис цілих розділів або таблиць.

- Очікування даних (найближчим часом): Delta Lake також підтримуватиме новий API для встановлення очікувань даних у таблицях або каталогах. Інженери зможуть вказати логічну умову і налаштувати серйозність для обробки очікуваних даних. Коли завдання Apache Spark записують дані в таблицю або каталог, Delta Lake автоматично перевіряє записи, а в разі порушення обробляє записи в залежності від наданої серйозності.

## 2.4 Архітектура Delta Lake

Архітектура Delta Lake розроблена для усунення необхідності завжди мати дворівневу архітектуру сховища даних / озера даних. При транзакціях ACID в озері даних базові файли даних, пов'язані з зовнішньою таблицею, не будуть оновлюватися до тих пір, поки транзакція або успішно не завершиться, або не завершиться повністю. Таким чином, це спрощує інкрементне завантаження (тобто додавання даних), уникаючи збоїв у процесі виконання завдань (тобто відновлення ETL) і зміни даних (тобто видалення/оновлення/вставки). В результаті розбіжності в метаданих значно скорочуються через відповідність вимогам ACID і контролю версій, які Delta Lake вносить в озера даних.[26]

Крім того, можливість комбінувати (тобто синхронізацію ETL) конвеєри пакетних і потокових даних забезпечує більшу узгодженість і відповідність для операцій в реальному часі та історичних відкатів [подорож у часі]. Тому застарівання дат зустрічається рідше, а історичні версії таблиць зберігаються, оскільки Delta Lake завжди створює новий журнал версій таблиць, доступний в режимі реального часу для кожної мікропакетної транзакції. Однією з найбільш очевидних проблем, пов'язаних з озерами даних, є досягнення стабільно високої продуктивності. Поширеним винуватцем є невеликі обсяги даних, розділені на [мільйони] занадто великої кількості файлів (тобто. Файли розміром в КБ). Реально, настройка робочих місць ETL може бути складною навіть для самих технологічно просунутих фахівців. Отже, Delta Lake може виконувати команди стиснення файлів для оптимізації

компонування і поділу даних для більш швидких результатів запитів і перевірки якості даних.

### 2.4.1 Внутрішні компоненти Delta Lake

Delta Lake містить деякі ключові Внутрішні компоненти, призначені для забезпечення надійності та узгодженості даних. Ними є:

1. Дельта-таблиці
2. Дельта-файли
3. Журнал різницевих транзакцій
4. Дельта-двигун
5. Дельта-рівень зберігання

Дельта-таблиці реєструються в метасторі (наприклад, в Apache Hive, AWS Glue) і містять базовий шлях до файлу даних, Властивості таблиці і визначення схеми. Дельта-файли зберігаються в хмарному сховищі об'єктів (наприклад, AWS, MS Azure, GCP) або файловій системі (наприклад, HDFS) у вигляді звичайних файлів даних (наприклад, Apache Parquet) або секціонованих папок (наприклад, Рік-місяць-День) в залежності від бізнес-логіки. Журнал Дельта-транзакцій-це дуже важлива папка з ім'ям `_delta_log`. По суті, це ядро і ключ до розуміння озера Дельта, тому що воно відстежує [по порядку] кожну виконувану транзакцію. Це єдине джерело достовірності і централізоване сховище змін дельта-таблиць. Наприклад, кожна атомарна фіксація в Дельта-таблиці створює новий файл JSON і файл CRC, що містить різні метадані і статистику таблиці. Цей механізм забезпечує транзакції ACID за допомогою атомарності, коли операції (наприклад, ВСТАВКА, оновлення) не будуть виконуватися в озері даних, якщо вони не будуть записані і повністю виконані в журналі транзакцій delta. Візуально, ось розбивка файлової системи високого рівня структури каталогів Delta Lake під капотом.

Delta lake поставляється в комплекті з механізмом delta і рівнем сховища delta, які є внутрішніми компонентами, інтегрованими за допомогою Apache Spark і хмарного сховища об'єктів. Движок забезпечує динамічне скорочення управління файлами,

дельта-кешування і безліч переваг оптимізації продуктивності. Рівень зберігання відстежує метадані, зміни схеми, управління версіями таблиць і багато операцій, що відповідають вимогам ACID.

## 2.4.2 Архітектура рішень Delta Lake

Delta lake слідує архітектурі шаблонів проектування, що забезпечує багаторівневу якість даних і контроль за допомогою номенклатури рівнів 3 таблиць.[22]. Вони є:

1. Бронза
2. Срібло
3. Золото

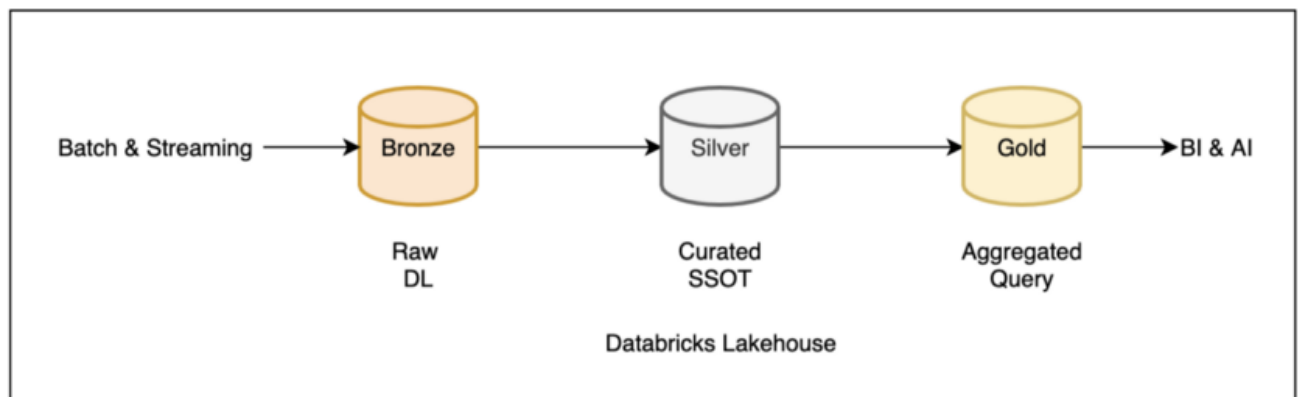


Рис. 1 Архітектура рівнів зберігання даних в Delta Lake

Кожен з цих рівнів служить важливою метою в конвеєрі архітектури delta, побудованому для забезпечення високої доступності даних для декількох наступних варіантів використання. Бронзовий рівень зберігає необроблені оперативні дані (тобто OLTP), зазвичай вимагає невеликої обробки даних і служить системою запису. Цей шар часто є зоною посадки, де Озеро даних приймає необроблені дані (наприклад, JSON, CSV) за допомогою комбінації пакетних і потокових завдань. Срібний рівень зберігає чисті атомарні дані (тобто OLAP), зазвичай включає в себе деяку обробку даних і служить єдиним джерелом істини з акцентом на дотримання схеми. Цей рівень може включати пакетні та потокові завдання, що перетворюють JSON в Parquet або ORC, а також перерозподіл даних відповідно до деякої бізнес-логіки і виконання методів стиснення файлів за допомогою Spark або Delta Lake для

підвищення продуктивності запитів. Золотий рівень зберігає агреговані вітрини ' відомчих даних (тобто OLAP), як правило, включає в себе багато обробки даних і служить в якості рівня представлення для бізнес-аналітики і запитів в галузі науки про дані. Цей рівень надає відфільтровані дані, готові для окремих користувачів виконувати спеціальні запити і використовувати Куби OLAP за допомогою інструментів BI, SQL, Python, Apache Spark і т. д.

Іноді Срібний шар може бути основним джерелом для машинного навчання та розробки функцій для створення сховища функцій, сховища кураторських даних або конвеєра виводу. Швидше за все, це буде залежати від організації та задіяних варіантів використання. Візуально, ось схема побудови конвеєра даних високого рівня для розширеної Дельта-архітектури Lakehouse.

Таким чином, Delta lake лідирує і впроваджує інновації в створенні єдиної платформи, яка може все це зробити. Це зменшує реальні проблеми переміщення та з'єднання даних між ізольованими системами (тобто аналогія → переміщення та зв'язок між окремими планетами в Сонячній системі) на підприємстві. Для того, щоб послідовно починати і приносити користь BI, AI / ML, аналізу великих даних тощо, я твердо вірю, що архітектура Lakehouse продовжуватиме розвиватися, навчати та розширювати можливості організацій для прийняття більш ефективних бізнес-рішень з використанням даних. Таким чином, такі проблеми, як "болото даних", брудні неетичні невідповідності даних, погана продуктивність вводу-виводу, повільне переміщення даних, проблеми з інтеграцією інструментів і командні Бункери, не будуть як і раніше заважати вам вирішувати проблеми клієнтів і надавати рішення, засновані на даних!

### 2.4.3 Архітектура з декількома переходами

Загальна архітектура використовує таблиці, що відповідають різним рівням якості в конвеєрі обробки даних, поступово додаючи структуру до даних: прийом даних (бронзові таблиці), проектування перетворень/функцій (срібні таблиці) і навчання машинному навчанню або прогнозування (золоті таблиці). У сукупності ми називаємо ці таблиці архітектурою з декількома переходами. Це дозволяє інженерам з обробки даних створювати конвеєр, який починається з необроблених даних як "єдиного джерела істини", з якого все випливає. Наступні перетворення та агрегації можуть бути перераховані та перевірені, щоб гарантувати, що Зведені таблиці бізнес-рівня все ще відображають базові дані, навіть якщо нижчі користувачі уточнюють дані та вводять контекстно-залежну структуру.

Варто трохи заглибитися в аналогію даних, щоб зрозуміти, як працює трубопровід озера Дельта (якщо ви дозволите нам розширений приклад). Замість планування серії окремих пакетних завдань для поетапного переміщення даних по конвеєру Delta Lake дозволяє даними текти як вода: плавно і постійно, в режимі реального часу.

Бронзові таблиці служать прототипом озера, в яке безперервно надходить величезна кількість води (даних). Коли він надходить, він брудний, тому що надходить з різних джерел, деякі з яких не такі чисті. Звідти дані постійно надходять в срібні таблиці, подібно верхів'ям струмка, з'єданого з озером, швидко рухається і постійно поточного. Оскільки вода (або дані, в нашому випадку) тече вниз по течії, вона очищається і фільтрується вигинами і поворотами річки, стаючи чистіше в міру її руху. До того часу, коли він досягає заводу з переробки води нижче за течією (наші золоті таблиці), він проходить деяку остаточну очистку і суворі випробування, щоб підготувати його до споживання, тому що споживачі (в даному випадку алгоритми ML) дуже прискіпливі і не будуть терпіти забруднену воду. Нарешті, з очисної установки він надходить в крани кожного наступного споживача (будь то алгоритми ML або аналітики BI), готовий до споживання в чистому вигляді.

Першим кроком у підготовці даних для машинного навчання є створення бронзової таблиці, місця, де дані можуть бути зібрані і збережені в максимально спрощеному вигляді.

Таблиці Delta Lake можуть легко обробляти безперервний потік даних як з історичних джерел, так і з джерел потокової передачі в реальному часі. І оскільки він використовує Spark, він майже універсально сумісний з різними форматами введення поточних даних і системами джерел, будь то Kafka, Kinesis, Cassandra або іншими.

## **2.5 Машинне навчання в Delta Lake**

Delta Lake також є платформою для життєвого циклу машинного навчання, оскільки вона пропонує інструменти та функції, які об'єднують науку про дані, розробку даних та виробничі робочі процеси, включаючи:

- Таблиці, які можуть безперервно обробляти нові потоки даних як з історичних джерел, так і з джерел потокової передачі в реальному часі, значно спрощуючи виробничий конвеєр data science.
- Застосування схеми, яка гарантує, що таблиці містяться в чистоті і порядку, не забруднюють стовпці і готові до машинного навчання.
- Еволюція схеми, яка дозволяє додавати нові стовпці в існуючі таблиці даних, навіть коли ці таблиці використовуються в робочому середовищі, не викликаючи критичних змін.
- Подорож у часі, також відома як управління версіями даних, дозволяє перевіряти, відтворювати або навіть відкочувати зміни в будь-якій таблиці озера Дельта, якщо це необхідно, у разі ненавмисних змін, внесених через помилку користувача.
- Інтеграція з MLflow, що дозволяє відстежувати і відтворювати експерименти шляхом автоматичної реєстрації експериментальних параметрів, результатів, моделей і графіків.

Ці особливості Delta Lake дозволяють інженерам з обробки даних та вченим розробляти надійні, стійкі, автоматизовані конвеєри даних та моделі машинного навчання швидше, ніж будь-коли.

## **2.6 Висновок**

У цьому розділі розглянуто методи обробки даних в режимі реального часу та методів інтеграції інформаційних систем і потоків даних. Розглянуто різні методи збору та обробки даних в режимі реального часу, розглянуто методи та випадки інтеграції потоків даних. Також розглянуто архітектуру Delta Lake бази даних для запису даних в архітектурі додатку, також розглянуто сховища які надає нам дана технологія, розглянута використання Delta Lake у різних випадках та внутрішні компоненти .

# **3 ПРИНЦИПИ РОБОТИ СИСТЕМИ EHEALTH ТА ДОСЛІДИТИ ВЗАЄМОДІЇ З API ХМАРНОЇ ПЛАТФОРМИ.**

## **3.1 Структура E-Health**

Термін "Електронна охорона здоров'я" - це аббревіатура, що позначає електронну охорону здоров'я. Всесвітня організація охорони здоров'я визначає "електронну охорону здоров'я" як використання інформаційних та комунікаційних технологій в інтересах охорони здоров'я. Однак не існує загальноприйнятого визначення електронної охорони здоров'я, і загальноприйнято включати безліч різних технологій та інструментів, також відомих як Інформаційні технології охорони здоров'я або Медична інформатика. Електронна охорона здоров'я використовує комп'ютерні технології, включаючи Інтернет та пов'язані з ним програми, для підвищення ефективності та результативності галузі охорони здоров'я.[23].

Серед технологій, включених у сферу електронної охорони здоров'я, - використання мобільних пристроїв в медичній допомозі( mHealth), таких як носяться пристрої, електронні щоденники або монітори дотримання; телемедицина, яка являє собою адаптацію інформаційних і мережевих технологій до клінічної практики та медичної освіти; платформи соціальних мереж та Інформатика охорони здоров'я, такі як електронні медичні записи та системи підтримки клінічних рішень. Нарешті, аналіз інформації, отриманої за допомогою цих інструментів, може бути виконаний з використанням технологій " великих даних " .

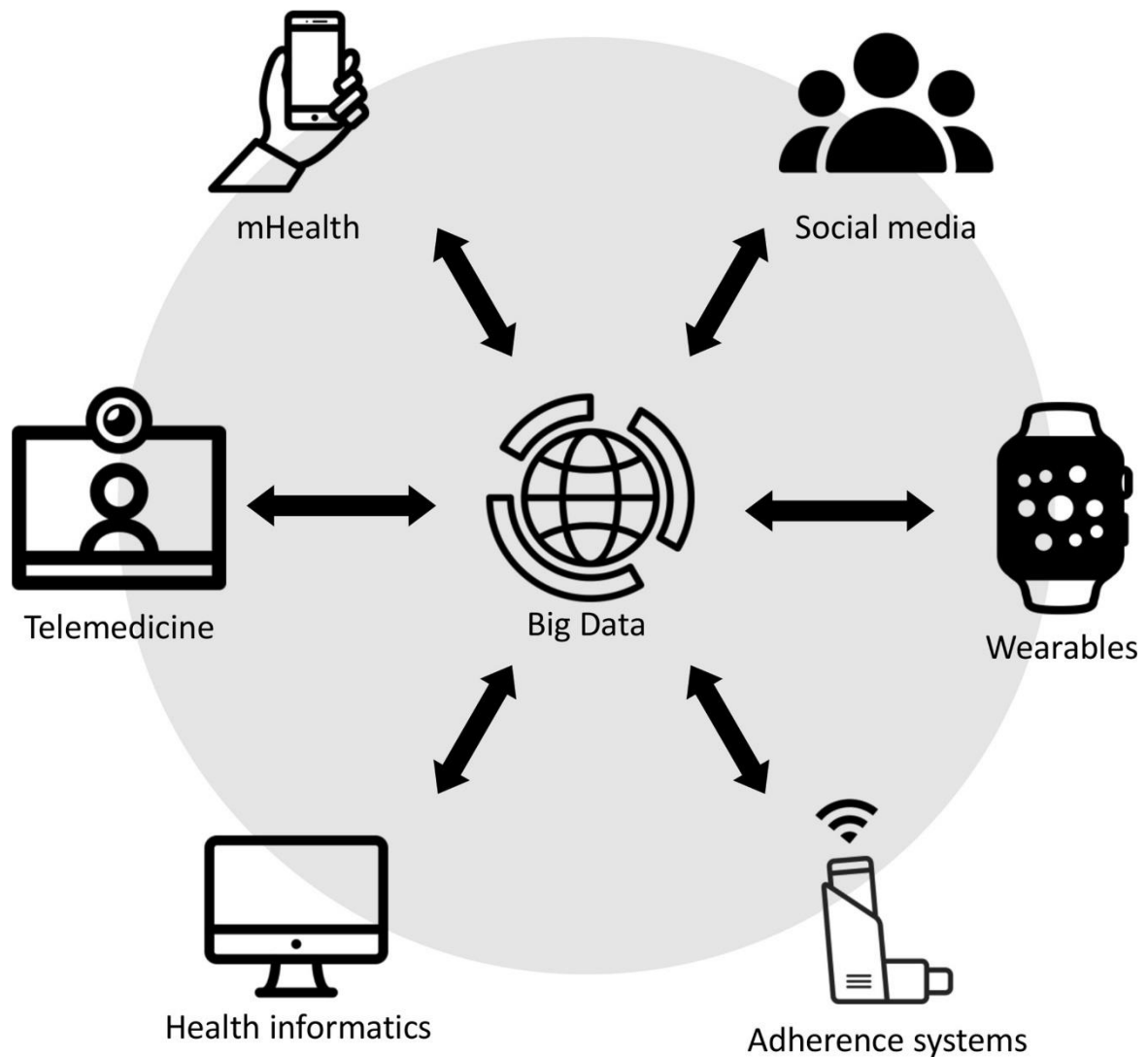


Рис.2 Взаємодія електронної охорони здоров'я з Big Data

Електронна охорона здоров'я включає в себе безліч технологій, які можуть бути застосовані в сфері охорони здоров'я. Інформація, отримана за допомогою цих інструментів, може бути використана в якості джерела великих даних, що, в свою чергу, може поліпшити систему.

Впровадження нових технологій в наше особисте життя різко зросло за останні кілька десятиліть. Персональні комп'ютери та ноутбуки призвели до появи більш легких пристроїв (наприклад, смартфонів або планшетів). Мобільний доступ в Інтернет став дешевше в розвинених країнах і, отже, широко поширений серед населення в цілому. У 2016 році, за оцінками, 75% європейських домогосподарств мали Портативне підключення до Інтернету.

Середовище охорони здоров'я не була чужа цій тенденції, але впровадження технологій йшло більш повільними темпами в порівнянні з особистим використанням пристроїв і додатків. Лікарі та пацієнти отримали можливість спілкуватися по-новому і збирати дані, які раніше були недоступні або які було дуже важко зібрати. Смартфони, додатки та інші так звані "розумні" пристрої разом з новими аналітичними інструментами відкрили нові горизонти для пацієнтів, клініцистів і дослідників. повідомляється, що пандемія коронавірусної хвороби 2019 року (COVID-19) за кілька місяців різко збільшила темпи впровадження комунікаційних технологій.

Впровадження різних технологій електронного охорони здоров'я вже було задокументовано для лікування різних хронічних захворювань, включаючи астму та алергічні стани. Однак є багато аспектів електронного здоров'я, які все ще залишаються невизначеними і знаходяться в стадії вивчення. У більшості випадків дані досліджень як і раніше обмежені, більш низької якості, засновані на поточних стандартах досліджень або непослідовні. як і раніше необхідно визначити і визначити заходи електронного охорони здоров'я або їх комбінації, які можуть бути адекватними для різних пацієнтів і алергічних розладів. Як і у випадку будь-якого клінічного втручання, деякі інструменти електронного охорони здоров'я можуть мати негативний вплив на певні вразливі підгрупи пацієнтів. Крім того, конфіденційність залишається важливою проблемою, яка завжди повинна вирішуватися з дотриманням найвищих стандартів захисту, як це передбачено конкретними урядовими постановами.

### **3.1.1 Електронна медична карта**

Системи електронних медичних карт набули широкого поширення в охороні здоров'я, що дозволяє генерувати великі обсяги реальних даних. Методи обробки природної мови-це підхід штучного інтелекту (AI), який витягує інформацію з клінічних описів в електронних медичних картах. Крім того, електронні медичні

карти дозволяють аналізувати тимчасові закономірності стану, які можуть виявити раніше невідомі зв'язки між діагнозами, тим самим надавши інформацію для майбутніх досліджень причинно-наслідкового зв'язку. Застосування цих технологій в практиці алергії все ще знаходиться на ранній стадії розробки. Існують дослідження, в яких розраховувалася харчова алергія або поширеність лікарської алергії в різних групах населення з використанням електронних медичних карт. Алгоритм обробки природної мови продемонстрував здатність визначати астму, аналізуючи такі записи. Потенційне використання AI в майбутньому може включати виявлення пацієнтів, які потребують лікування алергії, до того, як їх хвороба погіршиться. Це може бути особливо актуально для пацієнтів, які пропустили консультації або перервали лікування через пандемію COVID-19.

### **3.1.2 Телемедицина**

Телемедицина швидко отримує визнання як зручний спосіб доступу до медичних послуг у всьому світі. Тільки в 2019 році клініка Клівленда, велика система охорони здоров'я в Сполучених Штатах, здійснила 41000 віртуальних візитів, включаючи всі медичні спеціальності. За шість місяців 2020 року кількість віртуальних відвідувань підскочила до більш ніж 500000. 24 існують дослідження, які підтверджують, що телемедицина так само ефективна, як і особисті відвідування при амбулаторному лікуванні астми. Дослідження, в якому брали участь 169 дітей, які страждають на астму, порівнювали контроль над захворюванням і задоволеність пацієнтів після 6 місяців особистих відвідувань або телемедицини, і виявило, що вони схожі. Також опитування, які показали, що і пацієнти, і алергологи, як правило, не менше задоволені телемедициною або навіть віддають перевагу її особистим візитам. В результаті пандемії COVID-19 телемедицина стала невід'ємною частиною повсякденної практики клінік алергії та імунології в усьому світі. Телемедицина може бути особливо корисною для надання належної спеціалізованої допомоги при алергічних станах у сільських громадах, де особисті відвідування проблематичні через брак ресурсів. У таких країнах, як США, де приватна практика є звичайним

явищем, все ще існують значні політичні бар'єри, включаючи фрагментовану систему відшкодування витрат, які ускладнюють більш широке впровадження. Очікується, що в найближчому майбутньому телемедицина в поєднанні з введенням даних з датчиків, електронними медичними картами і штучним інтелектом для роботи з великими даними.

### **3.1.3 Абстракції E-Health**

#### **Рівень абстракції служби охорони здоров'я .**

Рівень абстракції послуг електронної охорони здоров'я містить набір послуг електронної охорони здоров'я та об'єднує дані та функціональні можливості, що містяться на рівні платформи електронної охорони здоров'я, з можливостями зв'язку на рівні платформи ІКТ.

#### **Рівень абстракції проміжного програмного забезпечення електронної охорони здоров'я.**

Абстракція проміжного програмного забезпечення електронної охорони здоров'я містить функціональні можливості, необхідні для роботи електронної охорони здоров'я незалежно від платформи ІКТ, використовуваної для передачі даних.

#### **Рівень абстракції підключення до електронної охорони здоров'я**

Рівень абстракції підключення містить технічні заходи для підключення компонентів електронної охорони здоров'я по мережі, в результаті чого створюється керований набір ресурсів ІКТ, що визначаються службою. Вимоги або можливості платформи ІКТ характеризуються як комбінація кожного з наступних класів Впровадження реальних телекомунікаційних послуг вимагає, щоб вони могли забезпечувати зіставлення з абстрактними послугами. Інтерфейс між рівнями в архітектурі повинен бути визначений в термінах цих абстракцій.

Примітка: кожен рівень буде пропонувати більше деталей, тобто менше абстракції в міру наближення до фактичної технології ІКТ зверху, і, відповідно, менше деталей у міру наближення до користувацького сервісу знизу.

Можливість підключення:

- режим підключення або безперервна швидкість передачі даних (CBR).
- Пакетний режим або змінна швидкість передачі даних (VBR).

#### Симетрія з'єднання:

- симетричний (тобто швидкість передачі даних, QoS, GoS, к/від користувача ідентична).
- Асиметричний (тобто швидкість передачі даних, QoS, GoS, к / від користувача не ідентична).

#### Топологія з'єднання (адреси):

- Одноадресна передача (тобто зв'язок направляється на одну кінцеву точку).
- Багатоадресна розсилка (т. е. зв'язок спрямована на окрему групу кінцевих точок).
- Широкомовна передача (тобто зв'язок направляється на всі кінцеві точки в мережі).

#### Тип контенту:

- Дані (тобто у вигляді файлів, наприклад, документа word, інформації про зображення, відкладеної передачі вибіркової інформації, такої як ЕКГ і відео і т.д.).
- Відео (наприклад, у формі трансляції в реальному часі і, в більш загальному плані, всі потоки даних в реальному часі, отримані з багатовимірних вибіркового сигналів в реальному часі).
- Аудіо (включаючи мову, ЕКГ і, в більш загальному плані, всі дані в реальному часі, отримані з одновимірних дискретизованих сигналів в реальному часі).
- Зображення (тобто. спеціалізована форма файлу даних, як правило, великого розміру, наприклад, Цифрова фотографія з високою роздільною здатністю), відкладена ЕКГ, відкладене відео і т. д.

### **3.1.4 Сумісність і працездатність**

Сумісність є основною метою системи електронної охорони здоров'я для забезпечення того, щоб Додатки могли підключатися і взаємодіяти один з одним, обмінюючись даними таким чином, щоб забезпечити дуже високий рівень гарантії продуктивності, безпеки, розуміння і дій. Архітектура, описана вище, є ключовим кроком у досягненні сумісності шляхом забезпечення загальних інтерфейсів між чітко визначеними площинами. Немає стимулів для забезпечення прямої сумісності на рівні платформи ІКТ, оскільки технології ІКТ призначені для ряду різних фізичних середовищ, тому в рамках електронної охорони здоров'я мета полягає в тому, щоб забезпечити сумісність самої платформи електронної охорони здоров'я і забезпечити, щоб ця платформа могла управляти доступними платформами ІКТ.

### **3.1.5 Особливості пристрою і системи**

Система охорони здоров'я заснована на передачі даних і спільному використанні пристроїв. Для забезпечення сумісності визначення даних повинні бути зрозумілі всіма пристроями з підтримкою електронної охорони здоров'я. Там, де використовуються радіопристрої, система електронної охорони здоров'я за замовчуванням вимагає, щоб такі пристрої могли працювати без перешкод і без обмежень по географічному розташуванню. Наприклад, імплантований датчик повинен бути здатний працювати у всіх областях і середовищах, в яких його власник може розумно бути присутнім, і, отже, повинна бути узгоджена здатність переносити і експлуатувати обладнання в областях, в яких користувач може носити обладнання (наприклад, турист, який подорожує з Лапландії в Гібралтар, розумно очікував би, що датчик буде працювати в обох місцях і не матиме ніякого негативного впливу на його здоров'я в будь-якому місці).

### **3.1.6 Безпека в електронній охороні здоров'я**

Термін "безпека" часто неправильно розуміють і частіше неправильно застосовують в телекомунікаціях. Пристрій стає безпечним, пропонуючи здатність бути несприйнятливим до деяких форм атак і коли воно представляє прийнятний і допустимий рівень ризику для користувача і системи, яка його оточує. У телекомунікаціях безпека часто представляється як здатність зберігати або підтверджувати ряд атрибутів користувача і їх зв'язку наступним чином:

- Справжність
- Повноваження
- Цілісність
- Конфіденційність

Крім того, в більшості випадків розумно додати до терміну безпеки інші вимірювання, такі як надійність і повторюваність, але такі речі часто виходять за рамки технічної стандартизації, хоча існує ряд посібників, в яких розглядаються такі питання, наприклад, ISO/IEC 27001

### **3.1.7 Конфіденційність**

Конфіденційність і захист конфіденційності - це непроста проблема безпеки, що вимагає розуміння взаємозв'язку даних з користувачами і потреб фахівців в області електронної охорони здоров'я. Всі користувачі електронної охорони здоров'я мають право на конфіденційність своїх даних, але важливо, щоб уповноважений медичний працівник мав доступ до медичних карт своїх пацієнтів. В основному це досягається за рахунок функції безпеки "повноваження", яка вимагає належної обережності при ідентифікації та аутентифікації.

Важливо визнати, що багато загроз конфіденційності можуть бути лише частково забезпечені технологією, в той час як основна частина засобів контролю конфіденційності носить організаційний і соціальний характер і залежить від ступеня довіри між постачальником даних (наприклад, пацієнтом) і одержувачем даних (наприклад, медичним працівником).

### 3.1.8 Абстрактна Модель Обслуговування Користувачів

Система електронної охорони здоров'я може бути спрощена до моделі збору необроблених даних, а потім їх передачі та обробки для отримання інформації про електронну охорону здоров'я.

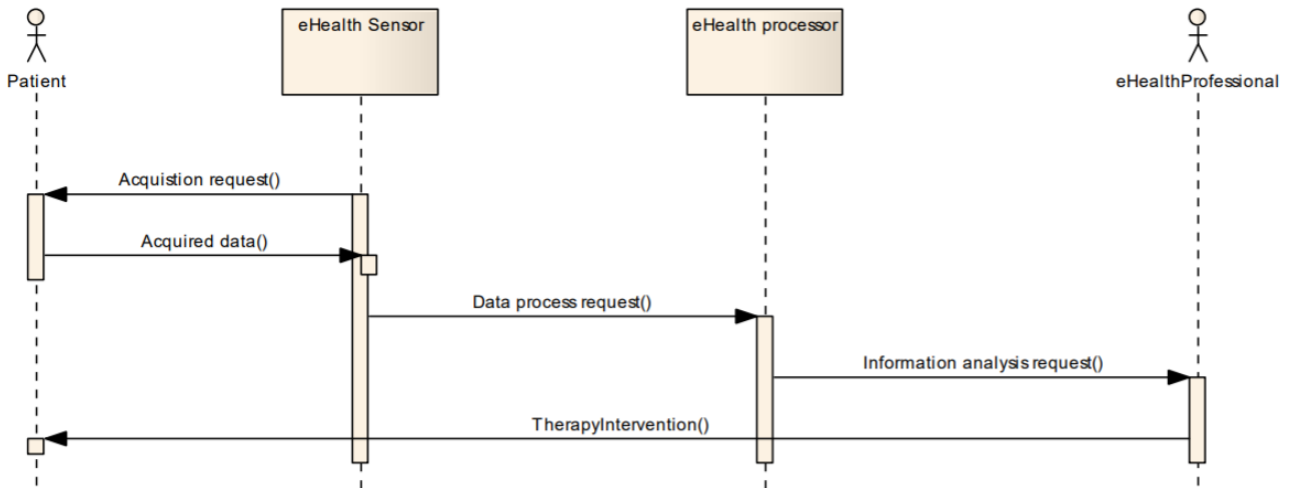


Рис.3 Абстрактна Модель Обслуговування Користувачів

#### Отримання інформації про електронну охорону здоров'я

В електронному охороні здоров'я важливим елементом успіху є збір даних, пов'язаних зі здоров'ям людини, званий фазою придбання. Хоча більша частина даних буде отримана в результаті прямого моніторингу пацієнта (за допомогою заборон, PAnS, PNs та PN-Fs), багато чого також буде отримано з знань про те, де живе пацієнт (дані про навколишнє середовище), історії здоров'я сім'ї (генетичні фактори) і що робить пацієнт (більше екологічних та контекстуальних факторів).

#### Обробка інформації про електронну охорону здоров'я

Необроблені дані часто є недостатньою основою для терапевтичного втручання, і тому інформація повинна бути отримана на основі цих даних. Ця дія з виконання або запуску інтелектуальних алгоритмів або логічних процедур з метою вилучення корисної інформації, яка може допомогти в процедурі прийняття рішення, тут називається обробкою. На цьому етапі ми спробуємо імітувати або полегшити завдання лікаря при прийнятті рішень.

Зіставлення інформації про електронну охорону здоров'я

Після вивчення та порівняння наявної інформації необхідно виділити розумну медичну інформацію або результати процесу. Зіставлення інформації про електронну охорону здоров'я об'єднує різні фрагменти даних, щоб можна було побачити подібності та відмінності (тобто автоматично відображати результати аналізу крові, встановлюючи логічні категорії, А також упорядковуючи дані на основі важливості).

### **Служби безпеки електронної охорони здоров'я**

Служби безпеки електронної охорони здоров'я відносяться до набору процедур, які забезпечують справжність, цілісність, конфіденційність і надійність при отриманні, обробці та використанні інформаційних процесів електронної охорони здоров'я, на додаток до забезпечення того, щоб тільки уповноважені сторони могли використовувати набір інформаційних послуг електронної охорони здоров'я. Крім того, Служби безпеки забезпечують захист учасників середовища електронної охорони здоров'я відповідно до існуючих правил (наприклад, конфіденційність, захист даних, правоохоронні органи).

### **Синхронізація інформації про електронну охорону здоров'я**

Акт оновлення інформації про електронну охорону здоров'я одного або декількох ресурсів таким чином, щоб вона вказувала або вказувала на одні й ті ж дані. Синхронізація дозволяє обмінюватися аналогічною інформацією в багатьох місцях і дозволяє уникнути ризику втрати даних при реплікації інформації. Це також може стосуватися акту оновлення інформації від особи, за якою ведеться спостереження, щоб продовжити термін дії.

### 3.1.9 Машинне навчання в E-Health

З розвитком технології глибокого навчання для обробки природної мови були застосовані моделі глибокого навчання, такі як циклічна нейронна мережа і згортова нейронна мережа [14]. Будь то дослідження електронної системи медичної документації або Інтелектуальний аналіз даних електронної медичної документації з використанням технології машинного навчання, пошук електронної медичної документації за кордоном, особливо в англійській електронній медичній документації, є зрілим [15, 16], що корисно для електронної охорони здоров'я. У цьому дослідженні здійсненість моделі перевіряється шляхом створення повністю пов'язаної моделі нейронної мережі для навчання; в той же час вона порівнюється з найвими методами класифікації листя і дерева рішень. У той же час, застосовуючи три способи, а саме, використовуючи технологію штучного інтелекту, не використовуючи технологію штучного інтелекту і використовуючи інші способи, Ми порівнюємо ефективність, зручність, стандарт та інші показники роботи з онлайн-кейсами. Нарешті, результати показують, що технологія штучного інтелекту може ефективно обробляти онлайн-інформацію про випадки захворювання для вирішення Електронної служби охорони здоров'я.

Метод класифікації методу глибокого навчання. Застосування методу глибокого навчання в класифікованих завданнях підвищить ефективність роботи. Наприклад, коли метод глибокого навчання застосовується до електронних випадків, він може ефективно призначати назви захворювань у випадках в категорію, яку зручно читати лікарям. Методи класифікації глибокого навчання включають повністю пов'язану нейронну мережу і циклічну нейронну мережу. У порівнянні з традиційними методами машинного навчання, методи глибокого навчання сильно відрізняються, і їх можна ефективно застосовувати для розпізнавання мови, класифікації тексту, розпізнавання зображень та інших завдань.

Контрольована навчальна програма. Контрольоване навчання та навчання під наглядом є основними завданнями навчання в теоретичній основі машинного навчання і є основою технології штучного інтелекту для роботи з онлайн-кейсами.

Контрольоване навчання-це зазначена навчальна задача, а загальна зазначена задача-задача класифікації. Для класифікації вхідні навчальні дані мають характеристики і мітки. Навчання полягає в тому, щоб знайти взаємозв'язок між функціями та мітками. Коли вводяться невідомі дані з об'єктами, але без міток, мітки невідомих даних відомі через існуючі зв'язки; тобто прогнозування здійснюється з урахуванням нових даних. У наведеній вище класифікації, якщо всі дані навчання позначені, це контрольоване навчання.

### **3.1.10 Особливості M-Health**

Інтернет речей (IoT), сучасний підхід, є перспективним рішенням у всьому світі для застосування в основному в галузі охорони здоров'я. Примітно, що IoT об'єднує майже 22 мільярди пристроїв і, як очікується, досягне 50 мільярдів пристроїв до 2030 року [1]. Інтернет Інтернету речей з'єднує цифрові пристрої і дозволяє обмінюватися інформацією між мережевими пристроями для вирішення певних проблем. Такі мережеві взаємодії, засновані на IoT, спростили життя людей. В даний час світ стикається з серйозними проблемами, пов'язаними зі здоров'ям [25]. Сектор охорони здоров'я, який використовує смартфони та пристрої на базі Інтернету речей, міг би допомогти пацієнтам контролювати свої повсякденні стани, пов'язані зі здоров'ям, а також в екстрених ситуаціях [25]. Отже, мобільна Охорона здоров'я (m-health) може забезпечити надання якісних медичних послуг у міру необхідності кожному пацієнту. Базуючись на IoT, технологія m-health може повністю забезпечити догляд за пацієнтом, і такі інноваційні технології адаптують обставини пацієнта, і їх показники можуть бути змінені в залежності від стану. Таким чином, в рамках Інтернету речей технологія m-health буде контролювати поточний і майбутній стан здоров'я тяжкохворих пацієнтів. Застосування Інтернету речей у програмах m-охорони здоров'я має високий потенціал для розширення можливостей систем первинної медико-санітарної допомоги, і, таким чином, було б легко забезпечити регулярний зв'язок між пацієнтами та медичними працівниками за участю лікарів. Наприклад, носійний пристрій може миттєво підключитися до інструменту для аналізу великих

даних, щоб безпосередньо ідентифікувати та обробляти будь-які коливання даних. Інший носійний датчик може перевіряти артеріальний тиск через шкіру, тоді як контактна лінза може досліджувати і показувати рівень цукру в сльозах . Останнім часом з'явилися програми комплексної оцінки здоров'я, засновані на телемедицині, проте все більше зростання інноваційних цифрових технологій охорони здоров'я пропонує нові рішення для своєчасного збору та обміну величезною кількістю даних про здоров'я. Таким чином, це забезпечує великий потенціал для розширення можливостей систем охорони здоров'я щодо поліпшення зв'язку між пацієнтом і медичними працівниками і зниження ризиків. Такі втручання могли б забезпечити всебічний догляд за пацієнтами, що знизило б ймовірність можливих ризиків. У медичних установах IoT заснований на наборах даних охорони здоров'я, підключенні по протоколу IP (IP), датчиках і клінічних пристроях, які можуть віддалено управляти електронними медичними записами. Об'єднання IP з корпоративними сервісними центрами та іншими клінічними пристроями дозволяє передавати дані між пацієнтами та медичними працівниками, які відомі як Інтернет здорових речей (ІОНТ). ІОНТ може надавати точні клінічні дані та ресурси охорони здоров'я в реальному часі для медичних працівників у життєво важливих ситуаціях, які потребують негайної допомоги. Конфігурації клінічних пристроїв стають все більш складними; тому вони кидають виклик експертам в області інформаційних технологій для включення m-health та Інтернету речей в системи охорони здоров'я.

Попит на дані для створення ефективної і добре структурованої системи m-охорони здоров'я, як правило, додає більше проблем. Отже, це рекомендується щодо інтелектуального бізнесу для розширення функціональності інтегрованих в IoT систем великих даних m-health. Цілі m-охорони здоров'я, засновані на цінності, стимулюють медичних працівників відповідно до надання своєчасної та доступної допомоги пацієнтам . Отримані результати, очевидно, свідчать про необхідну групову підтримку як медичних працівників, так і пацієнтів для ефективної роботи. Впровадження носяться пристроїв з технологією m-health і IoT є перспективним в плані підвищення доступності та доступу до недорогих медичних послуг. IoT

підкреслює універсальну мережу підключених до Інтернету послуг і пристроїв в будь-який час і в будь-якому місці, де IoT або інтернет пацієнтів (IOP) відображає доступні і прості комунікації між медичними працівниками і пацієнтами за допомогою захищеного підключення. У цьому розділі розглядаються програми Інтернету речей і m-health в охороні здоров'я. Вона також описує інноваційні методи поліпшення здоров'я за допомогою обчислювальних технологій та методів інтеграції Інтернету речей та M-охорони здоров'я, включаючи запропоновану модель для самостійного управління діабетом.

Для розміщення сервісів і даних була запропонована система моніторингу на основі m-health, в якій була запропонована інноваційна методика для організації та обробки онлайн-даних і управління додатками великих даних, інтегрованими з IoT, де розгорнута технологія отримувала дані пацієнтів з пристроїв IoT і успішно відправляла їх в хмару. Інший метод підкреслював перенесення отриманих даних m-health з загальних пристроїв Інтернету речей в хмару, де система управління набором даних дозволяла пристроям Інтернету речей своєчасно збирати життєво важливі дані споживачів. Хмара забезпечувала швидку обробку розміщених даних, тому споживач міг швидко отримувати оповіщення в критичних умовах. Інша модель була розроблена для обробки та відображення даних шляхом адекватного розгортання хмарної архітектури. В одній моделі застосовувалося шифрування для захисту обмінюваних клінічних даних. Іншим прикладом може служити визначення безпечних і надійних терміналів за участю лікарень m-health і медичних працівників для обміну клінічними даними шляхом встановлення зв'язку з пацієнтами. Кіберфізична структура, орієнтована на конкретного пацієнта, схвалила ефективні послуги mhealth шляхом систематичного збору даних у паралельних центрах обробки та розподілу житла. Цифрова система m-health була розроблена для дистанційного моніторингу пацієнтів з акцентом на прискорення підключення та обробки. Хмарна та електронна система моніторингу калорій, інтегрована в систему охорони здоров'я, яка точно класифікує різні продукти в їжі і сукупний вміст енергії в їжі. Була запропонована ефективна методика, заснована на принципах штучного інтелекту і

глибокого навчання, для прогнозування і точної класифікації частоти серцевих скорочень за допомогою вбудованого в смартфон датчика, званого акселерометром . Було запропоновано втручання m-health, засноване на хмарі Інтернету речей, для точного визначення голосу пацієнта, в якому використовується інтелектуальний інструмент моніторингу патології голосу на основі машинного навчання . Було описано ще одне ІОТ-хмарне втручання для моніторингу фізичної активності з використанням вбудованих клінічних датчиків, хмарної архітектури та веб-сервісів розширеної мови розмітки (XML) для простого, швидкого та безпечного вилучення та обміну даними, коли він повідомляє пацієнта про проблеми або відхилення під час фізичної активності . Принцип медичного промислового Інтернету речей (Healthiiot) був запропонований для своєчасного моніторингу інвалідів та геріатрії . Була запропонована модель контролю та моніторингу стану здоров'я, яка могла б здійснювати доступний і ефективний моніторинг пацієнтів із захворюваннями серця. Модель включає інтелектуальні шлюзи та енергоефективний датчик на основі Інтернету речей, де датчик може збирати дані про температуру тіла, рівні дихання та виведення електрокардіографії (ЕКГ); і відправляти ці дані на шлюзи для цифрового аналізу та інтерпретації в бездротовій мережі з незначними втратами . Було представлено втручання m-Health, яке могло б діяти як посередницький рівень між терміналами Інтернету речей і хмарою, де він застосовує брокера безпеки доступу до хмари (CASB) для підвищення конфіденційності та захисту даних . Була запропонована інша модель, яка могла б працювати з даними m-охорони здоров'я, чутливими до запізнювання . Була описана Сервіс-орієнтована структура для затвердження та вивчення необроблених клінічних даних, зібраних терміналами Інтернету речей за допомогою інтегрованих цифрових екземплярів з обмеженими ресурсами . Обговорювався інтелектуальний шлюз електронного охорони здоров'я, який міг би підтримувати послуги mhealth на основі Інтернету речей, де шлюзи знаходяться на різних вузлах розподіленої мережі, і кожен з них міг би працювати і управляти безліччю терміналів Інтернету речей, вільно пов'язаних як з медичними працівниками, так і з пацієнтами . Був описаний ще один інтелектуальний шлюз

електронного охорони здоров'я, який міг би забезпечити своєчасне розміщення, уточнення та аналіз даних про пацієнтів, а також оцінку раннього попередження на основі Інтернету речей (EWS) для тестування функціональності . Великі дані є ключовим елементом у m-health, оскільки клінічні датчики виробляють великі дані про здоров'я, тому був впроваджений масштабований датчик, вбудований в хмарну модель, який міг керувати всіма датчиками на основі m-health і своєчасно працювати з великими наборами даних . У контексті розумного міста особиста поведінка була досліджена саме шляхом застосування аналізу великих даних у соціальному IoT за допомогою підходів комплексного скринінгу.

### **3.2 Висновок**

У цьому розділі розглянуто основну архітектуру та структуру e-health, розглянуто технології які використовуються для телемедицини, особливості пристроїв та систем e-health, так саме розглянуто безпека в електронній охороні здоров'я та конфіденційність. Розглянуто моделі користування E-Health, машинне навчання в електронній охороні здоров'я та особистості m-health та використання його.

## 4 РЕАЛІЗАЦІЯ ДОДАТКА VAN З ІНТЕГРАЦІЄЮ ПОТОКІВ ДАНИХ В EHEALTH

З усіх розглянутих сервісів Amazon Web Services було обрано найкращі для вирішення нашої задачі. Архітектура нашого додатку буде виглядати так:

1. Дані які будуть надходити у додаток з пристроїв, збираються у Amazon Kinesis у режимі реального часу та буде розподіляти їх далі.
2. Після цього наступний сервіс зчитує дані з Amazon Kinesis та парсити дані до структурованого виду. Далі необроблені данні записуються в базу даних Delta Lake в перший рівень(Bronze). Після цього проводимо фільтрацію даних та записуємо в другий рівень нашої бази(Silver). Наступним кроком буде агрегація даних та запис у третій рівень бази(Gold).
3. Після аналізу даних записує їх у Amazon DynamoDB де користувачі та лікарі зможуть подивитися стан лідини за весь час роботи додатка.
4. Так само до нашої системи додається і пайплайн, який буде збирати дані за останню годину, робити репорт та виявляти відхилення пацієнта.

Ми можемо бачити на Рис.2 та Рис.3 як буде працювати додаток VAN, від надходження даних в Amazon Kinesis до запису у базу даних Amazon DynamoDB та вихід цих даних користувачу.

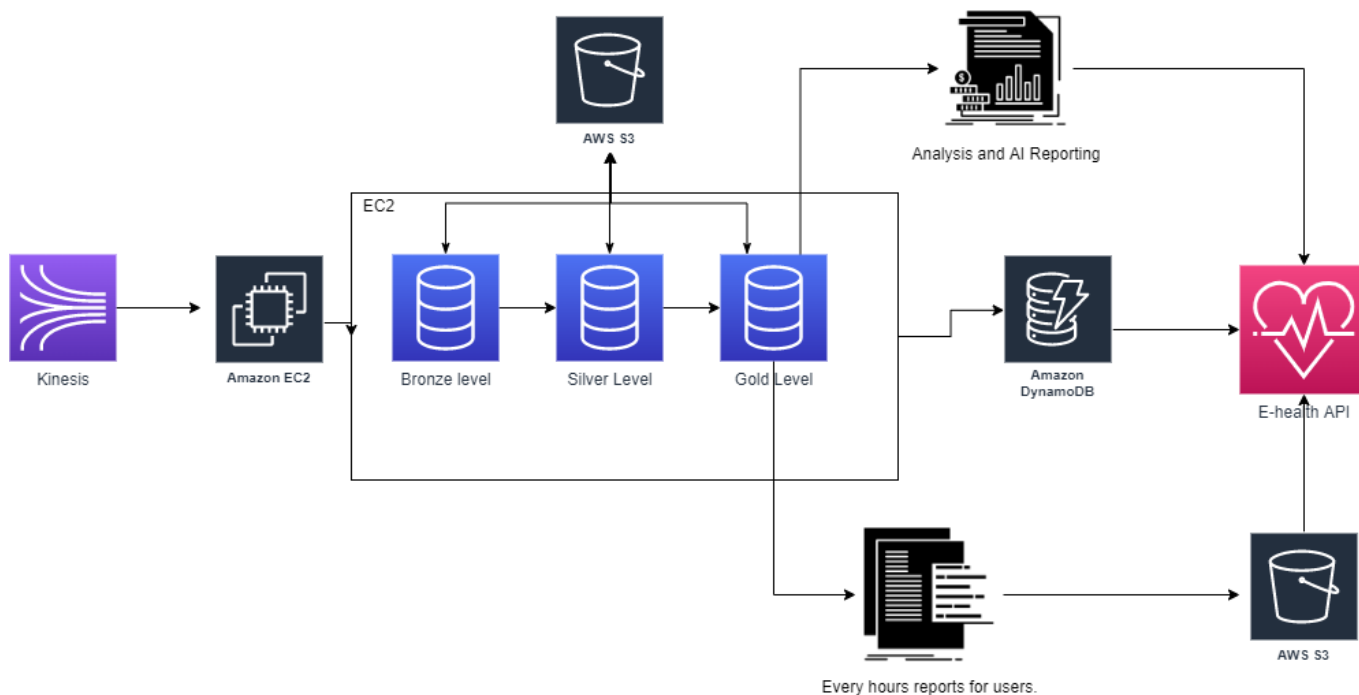


Рис. 4 Архітектура додатку BAN

#### 4.1 Збір інформації за допомогою Amazon Kinesis

Amazon Kinesis-це веб-сервіс Amazon, призначений для обробки великомасштабних потоків даних з безлічі сервісів в режимі реального часу. Його можна розглядати, як і Apache Kafka, як свого роду брокер повідомлень. Це означає, що він діє як посередник між різними джерелами генерації даних, дозволяючи іншим додаткам або службам працювати з вихідними даними.

Однією з ключових переваг Kinesis є те, що він дозволяє обробляти і аналізувати дані практично негайно, замість того, щоб чекати надходження повного набору даних, потім обробляти його, а потім відправляти для аналізу. Розуміння може бути отримано за хвилини, а не за години, дні або тижні. Kinesis робить це можливим без тижнів складної настройки, оскільки він поставляється у вигляді керованої платформи, а це означає, що вам не потрібно керувати будь-якою інфраструктурою. Amazon Kinesis data Streams (KDS) розроблений як масштабований і стійкий сервіс потокової передачі даних в реальному часі. KDS використовується, коли у вас є великий обсяг потокової передачі даних від безлічі потенційно нетрадиційних виробників даних. Він може приймати гігабайти даних в секунду з безлічі джерел,

включаючи (але не обмежуючись ними) кліки по веб-сайтам, потоки подій бази даних, фінансові транзакції, ігрові мікротранзакції, пристрої Інтернету речей і події відстеження місця розташування.

Іншими словами, якщо дані, які ви хочете передати в потоковому режимі, повинні надходити безпосередньо і бути доступні для використання службою або додатком, або повинні бути проаналізовані відразу після отримання, KDS-це варіант для вас. Зібрані дані майже відразу ж стають доступними (протягом 70 мілісекунд після збору) для аналітики в режимі реального часу, дозволяючи створювати інформаційні панелі в режимі реального часу, виявляти аномалії в режимі реального часу, динамічне ціноутворення і т. д.

Kinesis замінює традиційні посередники повідомлень в середовищах, які приймають великі потоки даних, які необхідно обробити і доставити в інші додатки і служби.

Amazon Kinesis - це керований сервіс, який вимагає мінімального налаштування та налаштування. Ключові поняття, що використовуються Kinesis, включають виробників даних, споживачів даних, Потоки даних, сегменти, Записи даних, Ключ розділу і порядковий номер.

Виробники даних-це пристрої-джерела, які передають записи даних. Споживач даних витягує Записи даних із сегментів у потоці. Споживач-це додаток або послуга, які використовують потокові дані. Сегменти складаються з цих записів даних, і, у свою чергу, численні сегменти складають потік даних Kinesis. Ключ розділу-це ідентифікатор, такий як ідентифікатор користувача або відмітка часу, при цьому порядковий номер служить унікальним ідентифікатором для кожного запису даних. Це гарантує, що дані залишаються незмінними протягом усього потоку.

Kinesis може підтримувати Android, Java, Go та .NET, серед іншого. Kinesis в даний час може запропонувати більшу гнучкість в інтеграції, він менш гнучкий з точки зору конфігурації – він дозволяє налаштовувати тільки кількість днів і сегментів і виконувати Синхронний запис на 3 різних машини, центри обробки даних і зони доступності (ця стандартна конфігурація може обмежити продуктивність). масштабованою і забезпечувати більшу пропускну здатність.

Недолік гнучкості конфігурації Kinesis обумовлений його конструкцією. Стандартизована конфігурація - це частина того, що дозволяє налаштувати її за години, а не за тижні. Саме тому Kinesis пропонує окремі рішення, такі як пожежний шланг, потокове відео, аналіз даних та потокове передавання даних. Ці конфігурації з урахуванням конкретних варіантів використання дозволяють використовувати Kinesis в більшій кількості сценаріїв, зберігаючи при цьому переваги керованого рішення з швидким налаштуванням.

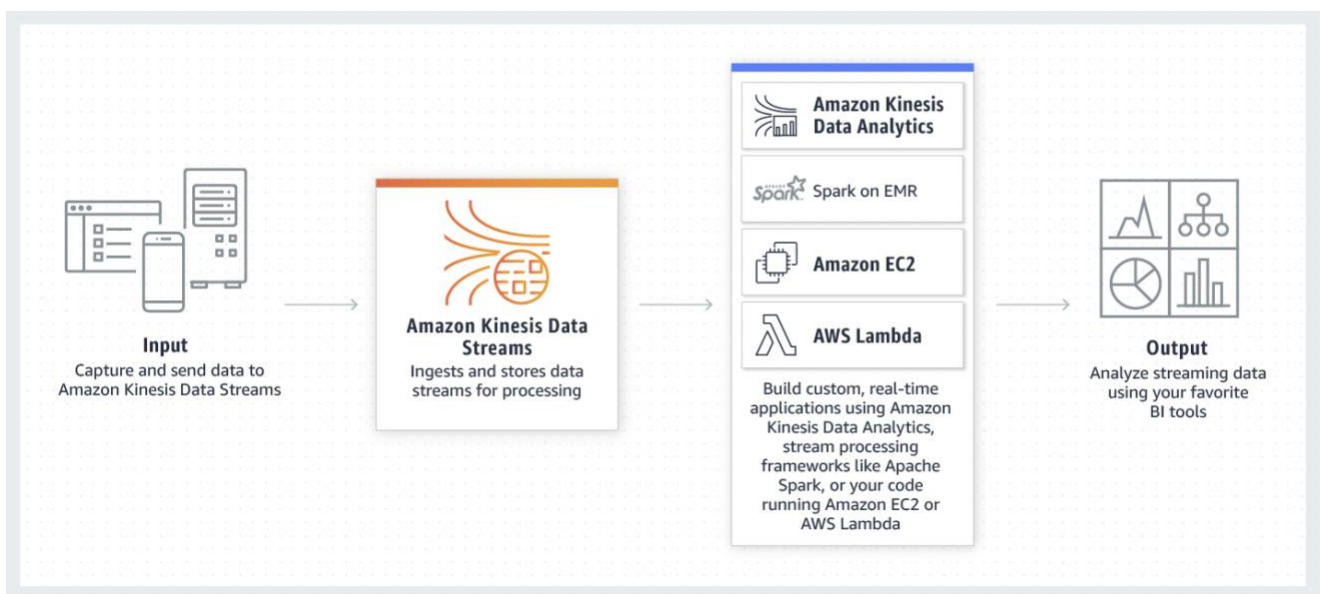


Рис.5 Amazon Kinesis Data Streams.[17].

У сервісі Kinesis Data Streams створили додаток потокової передачі під назвою “data-stream” для завантаження та обробки інформації яка поступає з сенсорів. Данні будуть поступати до нашого конвеєра даних у режимі реального часу. На рис. 5 можемо побачити які характеристики має наш Kinesis Data Streams.

## Create a data stream [Info](#)


### Data stream configuration

Data stream name

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.

### Data stream capacity [Info](#)

Data records are stored in Kinesis Data Stream. A shard is a uniquely identified sequence of data records in a stream.

[Request limit increase](#) 

### ▼ Shard estimator

Use the shard estimator to generate the recommended optimal number of shards for your stream based on your usage.

#### Writing to the stream

Average record size (in KiB)

Minimum: 1 KiB, maximum: 1024 KiB.

Maximum records written per second

#### Reading from the stream

Total number of consumers

Рис.6 Створення потоку Kinesis

Також на рис. 6 можемо побачити код написаний на мові програмування Python який буде надсилати інформацію з датчиків на наш конвеєр даних, кожну секунда. Таким чином ми імітуємо надсилання датчиком інформацію яку він отримав від користувача.

```

}class KinesisLoader:
}    def __init__(self, batch_size=20, maximum_records=21):
}        """
}        The default batch_size here is to match the maximum allowed by Kinesis in a PutRecords request
}        """
}        self.batch_size = min(batch_size, 500)
}        self.maximum_records = maximum_records
}        self.kinesis_client = boto3.session.Session().client('kinesis', 'us-west-2')
}
}    def generate_and_submit(self):
}        counter = 0
}        # Simple cutoff here - guaranteed to not send in more than maximum_records, with single batch granularity
}        while counter < self.maximum_records and counter < self.maximum_records - self.batch_size:
}            records_batch = [self.get_kinesis_record() for _ in range(0, self.batch_size)]
}            request = {
}                'Records': records_batch,
}                'StreamName': 'data-stream'
}            }
}
}            response = self.kinesis_client.put_records(**request)
}            self.submit_batch_until_successful(records_batch, response)
}
}            counter += len(records_batch)
}            print('Batch inserted. Total records: {}'.format(str(counter)))
}
}        return
}
}    def submit_batch_until_successful(self, batch, response):
}        """ If needed, retry a batch of records, backing off exponentially until it goes through"""
}
}        retry_interval = 0.5
}
}        failed_record_count = response['FailedRecordCount']
}        while failed_record_count:
}            time.sleep(retry_interval)
}
}            # Failed records don't contain the original contents - we have to correlate with the input by position
}            failed_records = [batch[i] for i, record in enumerate(response['Records']) if 'ErrorCode' in record]
}
}            print('Incrementing exponential back off and retrying {} failed records'.format(str(len(failed_records))))
}            retry_interval = min(retry_interval * 2, 10)
}            request = {
}                'Records': failed_records,
}                'StreamName': 'data-stream'
}            }
}
}

```

Рис.7 код на мові програмування Python для відправки даних Kinesis Data Streams

На Рис. 7 ми можемо побачити діаграму яка показує що данні успішно отримуються та обробляються нашим Kinesis Data Streams. З діаграм ми можемо побачити кількість отриманих запитів у байтах ,у кількості та в мілісекундах, також який процент прийнятих даних отримано успішно.

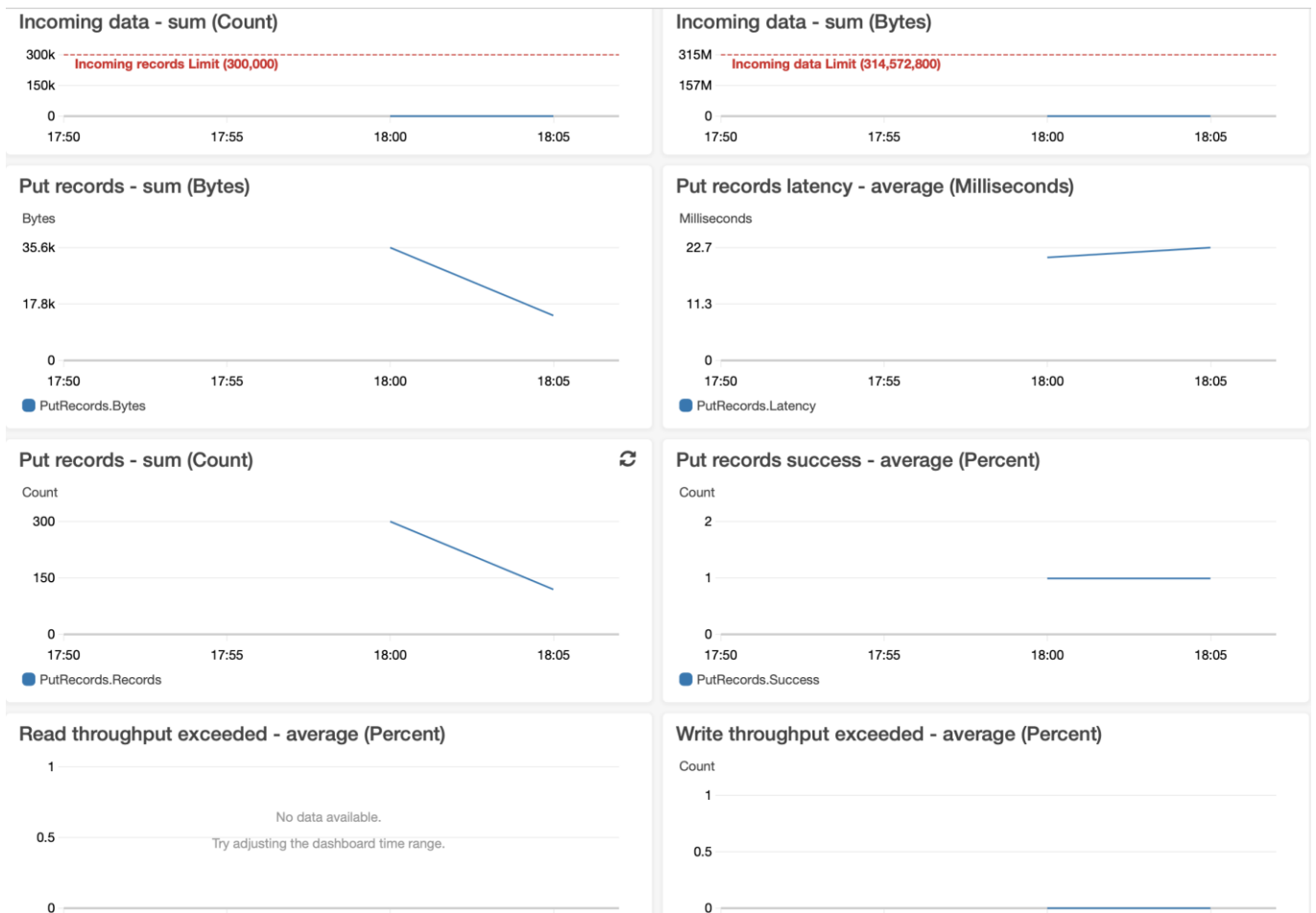


Рис.8 Діаграма роботи Kinesis Data Streams

## 4.2 Обробка та аналіз даних використовуючи Delta Lake у EC2

Обчислювальна хмара Amazon Elastic Compute Cloud – Amazon EC2)–це веб-сервіс, що надає безпечні масштабовані обчислювальні ресурси в хмарі. Він допомагає розробникам, спрощуючи проведення обчислень в хмарі в масштабі всього інтернету. Простий веб-інтерфейс сервісу Amazon EC2 дозволяє отримати доступ до обчислювальних ресурсів і налаштувати їх з мінімальними зусиллями. Він надає користувачам повний контроль над обчислювальними ресурсами, а також перевірену обчислювальну середу Amazon Для роботи.

Amazon EC2 пропонує обчислювальну платформу з найбільш широкими і доскональними функціональними можливостями, яка дозволяє вибрати процесор,

сховище, мережу, операційну систему і модель покупки. Ми пропонуємо найшвидші процесори в хмарі і є єдиним хмарним простором з мережею Ethernet, пропускна здатність якої становить 400 Гбіт/с. Маємо в своєму розпорядженні найпотужніші інстанси на базі графічних процесорів для організації курсів з машинного навчання і графічного відображення робочих навантажень, а також інстанс з найнижчими витратами на кожен логічний висновок інстансов в хмару. На AWS працює більше робочих навантажень SAP, HPC, машинного навчання та Windows, ніж у будь-якій іншій хмарі.

Amazon EC2 пропонує найширший вибір різноманітних інстанцій, побудованих на основі новітніх технологій сховищ, мереж і обчислень і розроблених для забезпечення високої продуктивності і безпеки.

## Прискорення впровадження інновацій та підвищення рівня безпеки з AWS Nitro System

AWS Nitro System-базова платформа для нового покоління універсальних віртуальних машин EC2, що дозволяють використовувати виділене обладнання та програмне забезпечення для виконання багатьох традиційних завдань віртуалізації, щоб підвищити продуктивність, доступність і безпеку при одночасному скороченні витрат, пов'язаних з віртуалізацією. Nitro System пропонує широкий вибір структурних блоків, які можна компонувати різними способами, що забезпечує гнучкі можливості для проектування і високу швидкість доставки нових типів інстанцій EC2 з постійно розширюється вибором варіантів обчислювальних ресурсів, сховищ, пам'яті і мережевих конфігурацій.

## Вибір процесорів

Вибір з процесорів останнього покоління Intel Xeon, AMD EPYC і AWS Graviton дозволяє найкращим чином збалансувати продуктивність і вартість ваших інструментів. Інстанси EC2 на базі NVIDIA GPUs і AWS Inferentia доступні в тому

числі для інструментів, які вимагають прискорених обчислень, таких як машинне навчання, ігри та додатки з інтенсивним використанням графіки.

### Високопродуктивне сховище

Amazon Elastic Block Store (EBS) пропонує простий у використанні, високопродуктивний сервіс блочного сховища, створений для використання з Amazon EC2. Amazon EBS надається в широкому діапазоні типів томів, який забезпечує вибір оптимально підходящих для робочого навантаження рівнів продуктивності сховища і вартості. Багато типів інстанцій EC2 також йдуть в комплекті з варіантами локальних сховищ NVMe SSD для додатків, що вимагають низьких затримок.

### Покращена Мережева конфігурація

AWS-перша і єдина хмара, що пропонує для обчислення інстанцій поліпшену мережеву конфігурацію Ethernet з пропускну здатністю мережі до 400 Гбіт/с. завдяки поліпшеній мережевої конфігурації досягається значно вища продуктивність у вигляді кількості переданих пакетів в секунду (PPS), а також знижуються нестійкість і затримки мережі. Для додатків, пов'язаних з високопродуктивними обчисленнями (HPC), Elastic Fabric Adapter являє собою мережевий інтерфейс для інстанцій Amazon EC2, який забезпечує низьку затримку, високу пропускну здатність з'єднань між обчислювальними вузлами і можливість масштабування додатків до декількох тисяч ядер.

### Вибір моделі покупки

Ми пропонуємо кілька моделей покупки на вибір: на вимогу, спотові інстанси і Savings Plan. Спотові інстанси дозволяють заощадити до 90 % на відмовостійких робочих навантаженнях. Savings Plan дозволяють заощадити до 72 % завдяки використанню відповідно до зобов'язань і гнучкості в рамках EC2, Fargate і Lambda. Ви також можете оптимізувати свої витрати за допомогою рекомендацій по

інстанціях, інтегрованих в EC2, за допомогою AWS Compute Optimizer або за допомогою таких інструментів, як cost Explorer.

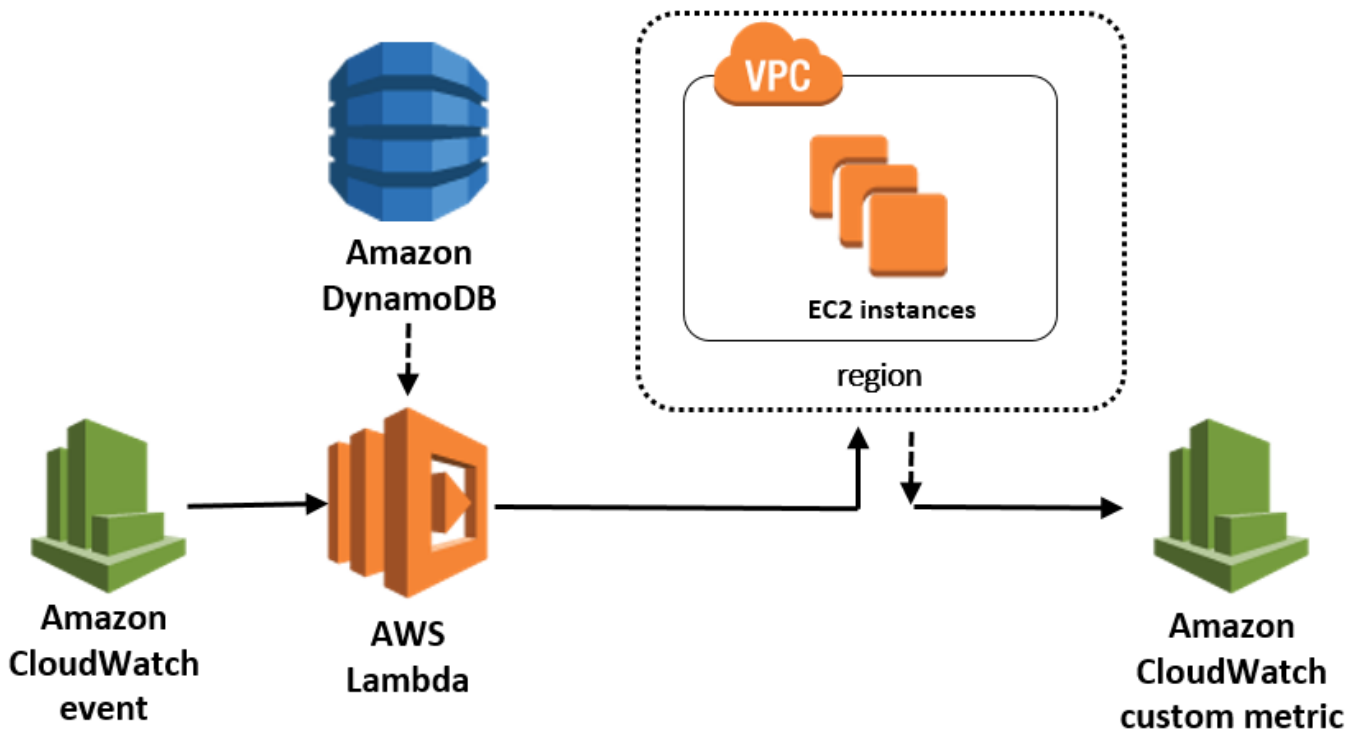


Рис.9 Принцип роботи сервісу AWS EC2[18].

Amazon Elastic Compute Cloud (Amazon EC2) забезпечує масштабовані обчислювальні потужності в хмарі Amazon Web Services (AWS). Використання Amazon EC2 позбавляє вас від необхідності інвестувати в обладнання заздалегідь, тому Ви можете швидше розробляти і розгортати додатки. За допомогою Amazon EC2 ви можете запускати стільки або стільки віртуальних серверів, скільки вам потрібно, налаштовувати безпеку і Мережу, а також управляти сховищем. Amazon EC2 дозволяє збільшувати або зменшувати масштаб для обробки змін вимог або сплесків популярності, зменшуючи необхідність прогнозування трафіку.

Amazon EC2 надає наступні функції:

- Віртуальні обчислювальні середовища, відомі як екземпляри
- Попередньо налаштовані шаблони для ваших екземплярів, відомі як зображення машин Amazon (AMI), які упаковують біти, необхідні для вашого сервера (включаючи операційну систему та додаткове програмне забезпечення)

- Різні конфігурації процесора, пам'яті, сховища та мережевої ємності для ваших екземплярів, відомі як типи екземплярів
- Захищена інформація для входу в ваші екземпляри за допомогою пар ключів (AWS зберігає відкритий ключ, а ви зберігаєте закритий ключ в безпечному місці)
- Томи сховища для тимчасових даних, які видаляються при зупинці, переході в сплячий режим або завершенні роботи екземпляра, відомі як томи сховища екземплярів
- Томи постійного сховища для ваших даних за допомогою Amazon Elastic Block Store (Amazon EBS), відомого як томи Amazon EBS
- Кілька фізичних місць розташування для ваших ресурсів, таких як екземпляри і томи Amazon EBS, відомі як регіони і зони доступності
- Брандмауер, що дозволяє вказувати протоколи, порти і діапазони IP-адрес джерел, які можуть досягати ваших екземплярів за допомогою груп безпеки
- Статичні IPv4-адреси для динамічних хмарних обчислень, відомі як еластичні IP-адреси
- Метадані, відомі як теги, які ви можете створювати та призначати своїм ресурсам Amazon EC2
- Віртуальні мережі, які ви можете створювати, логічно ізольовані від решти Хмари AWS і які ви можете при необхідності підключати до своєї власної мережі, відомої як віртуальні приватні Хмари (VPC).

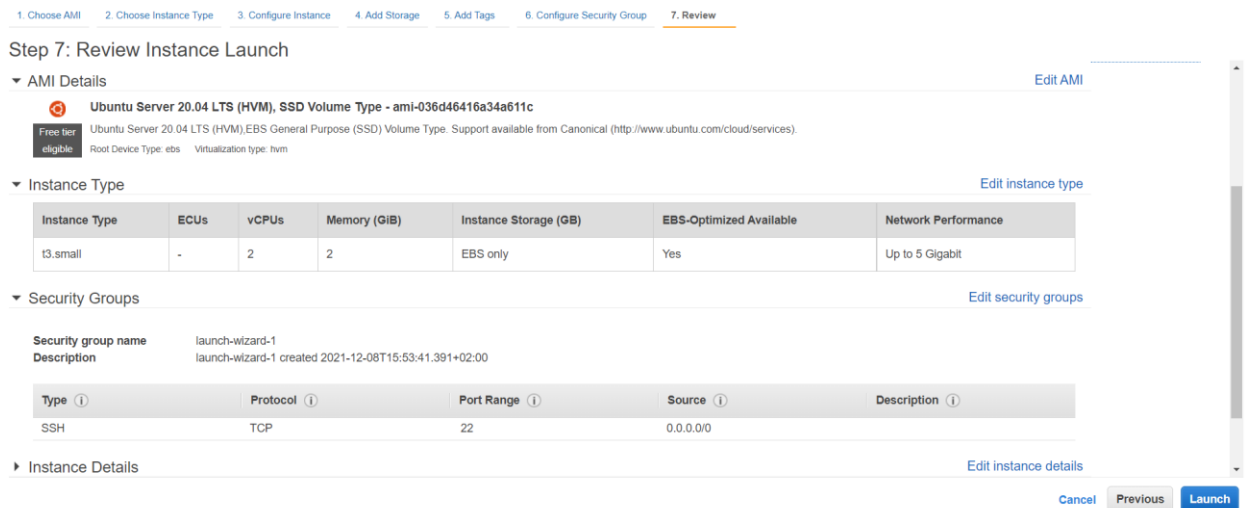


Рис.10 Створення EC2

Алгоритм роботи :

Після того як Kinesis Data отримує дані користувача Delta Lake починає збирати дані пацієнта та зберігає у першому рівні бази – бронзовому рівні. Данні які отримує Delta Lake надходять у JSON форматі та мають такий вигляд:

```
{
  "bodyTemperature": 36.6,
  "sugar": 4.0,
  "upperPressure": 120,
  "lowerPressure": 90,
  "hemoglobin": 145,
  "actionData": {
    "avg_rss12": [30.00],
    "var_rss12": [0.00],
    "avg_rss13": [5.00],
    "var_rss13": [2.35],
```

```
"avg_rss23": [7.75],  
"var_rss23": [1.92]  
}  
}
```

З яких ми можемо побачити що ми отримуємо:

bodyTemperature – температура тіла користувача,

sugar – зміст цукру у крові користувача,

upperPressure – верхній показник тиску користувача,

lowerPressure – нижній показник тиску користувача,

hemoglobin – гемоглобін користувача,

avg\_rssXY та var\_rssXY – дані с датчиків руху та положення тіла користувача;

Після збереження необроблених даних на перши рівень, ми проводимо фільтрацію даних та записуємо їх записуємо в другий рівень нашої бази – срібний рівень.

Наступним кроком буде агрегація даних та запис у третій рівень бази – золотий рівень

. Після цього проводимо аналіз оброблених даних на виявлення відхилень.

Для отримання результатів аналізу ми дивимося на лікарняну картку яка є у нашій базі даних. У цій карті записано кількість років користувача, чи є в нього хронічні захворювання, наприклад діабет, астма, чи були проблеми з роботом серця та інші.

Для обробки частини даних ми скористалися машинним навчанням методом кластерезації. Для цього ми навчили нашу програму на відкритому датасеті з інформацією по руху людини та положенню його тіла. На рис. 12 можемо побачити код програми на мові програмування Python навчання та використання його в нашій лямбда функції. Після аналізу даних ми можемо виявити в якому положенні та що за дію виконує користувач. Список дій користувача: біжить, стоїть, лежить, нагнувся, присів, ходить.

```

)import joblib
import pandas as pd
from pandas.io.json import json_normalize

from sklearn import svm
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
)from sklearn.neighbors import KNeighborsClassifier

)class Action:

)    def __init__(self):
        joblibKNN_file = "KNN_model.pkl"
        joblibSVC_file = "SVC_model.pkl"
        self.KNN_model = joblib.load(joblibKNN_file)
)        self.SVC_model = joblib.load(joblibSVC_file)

)    def classifierAction(self, dataAction):
        actionData = pd.DataFrame.from_dict(dataAction)
)        return self.KNN_model.predict(actionData)

data = pd.read_csv('/Users/ekaterina/PycharmProjects/BAN/events/all.csv')
X = data.iloc[:, :-1].values
y = data['action']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=27)

SVC_model = svm.SVC()
KNN_model = KNeighborsClassifier(n_neighbors=9)

SVC_model.fit(X_train, y_train)
KNN_model.fit(X_train, y_train)
SVC_prediction = SVC_model.predict(X_test)
KNN_prediction = KNN_model.predict(X_test)
print(accuracy_score(SVC_prediction, y_test))
print(accuracy_score(KNN_prediction, y_test))
print(confusion_matrix(SVC_prediction, y_test))
print(classification_report(KNN_prediction, y_test))
print(y_test)
testdata = pd.read_csv('/Users/ekaterina/PycharmProjects/BAN/events/testII.csv')
print(SVC_model.predict(testdata))
print(KNN_model.predict(testdata))

```

Рис.11 Код програми на мові програмування Python Навчання та використання штучного інтелекту

Після того як ми отримали яку дію та в якому положення зараз знаходиться користувач ми можемо проводити аналіз його показників тіла. Майже кожен з показників тіла буде змінюватися від дій користувача, таким чином ми обробили ці дані для кожної дії та для кожного положення тіла. Тобто якщо людина була у стані спокою та її серце биття було у нормі та через мить людина почала бігти то биття серця почало зростати але у додаток не виявить ніяких відхилень.

Також якщо у людини є хронічні захворювання то для неї норма показників тіла будуть відрізнятися від показників здорової людини, що ми також повинні врахувати.

```
import json
```

```
class Norm:
    def __init__(self, json):
        with open(json) as json_file:
            norm = json.load(json_file)
            self.bodyTemperature = norm["bodyTemperature"]
            self.sugar = norm["sugar"]
            self.upperPressure = norm["upperPressure"]
            self.lowerPressure = norm["lowerPressure"]
            self.hemoglobin = norm["hemoglobin"]
            self.criticalBodyTemperature = norm["criticalBodyTemperature"]
            self.criticalSugar = norm["criticalSugar"]
            self.criticalUpperPressure = norm["criticalUpperPressure"]
            self.criticalLowerPressure = norm["criticalLowerPressure"]
            self.criticalHemoglobin = norm["criticalHemoglobin"]
```

Рис. 12 Аналіз даних обробки стану користувача

Якщо при аналізі даних користувача буде виявлене не критичне відхилення показників тіла, то користувача додаток проінформує на пошту що треба звернутися до лікаря та здати аналізи на наявність захворювання. Якщо показники тіла користувача будуть у критичному стані додаток проінформує користувача та викличе швидку допомогу на місце знаходження користувача через інформацію з датчика.

### 4.3 Збереження Delta Lake даних на S3

Amazon Simple Storage Service ( Amazon S3) - це служба зберігання об'єктів, яка забезпечує кращу в галузі масштабованість, доступність даних, безпеку і продуктивність. Клієнти всіх розмірів і галузей можуть використовувати Amazon S3 для зберігання і захисту будь-якого обсягу даних в різних випадках використання, таких як озера даних, веб-сайти, мобільні додатки, резервне копіювання і відновлення, архівування, корпоративні додатки, пристрої Інтернету речей і аналіз великих даних. Amazon S3 надає функції управління, що дозволяють оптимізувати, організувати та налаштувати доступ до даних відповідно до конкретних бізнес-вимог, вимог організації та вимог відповідності.

Amazon S3-це служба зберігання об'єктів, яка зберігає дані у вигляді об'єктів у кошиках. Об'єкт-це файл і будь-які метадані, що описують файл. Відро-це контейнер для предметів.

Щоб зберігати свої дані в Amazon S3, спочатку створіть кошик і вкажіть ім'я кошика і Регіон AWS. Потім ви завантажуєте свої дані в цю корзину як об'єкти в Amazon S3. У кожного об'єкта є ключ (або ім'я ключа), який є унікальним ідентифікатором об'єкта в кошику.

S3 надає функції, які ви можете налаштувати для підтримки вашого конкретного варіанту використання. Наприклад, ви можете використовувати керування версіями S3 для зберігання декількох версій об'єкта в одному кошику, що дозволяє відновлювати випадково видалені або перезаписані об'єкти.

Кошки та об'єкти в них є приватними і доступні тільки в тому випадку, якщо ви явно надасте дозволи на доступ. Для управління доступом можна використовувати політики кошика, політики управління ідентифікацією і доступом AWS (IAM), списки управління доступом (ACL) і точки доступу S3.

Робота додатку з сервісом AWS S3:

Delta Lake зберігає дані в базі в структурованому форматі в parquet файлах, для кожного рівня збереження даних. Тому при записі збереження кожного рівня будуть

зберігатися в parquet файлах для кожного користувача окремо по їх особистими ідентифікаціями в системі.

#### **4.4 Збереження даних у Amazon DynamoDB**

Amazon DynamoDB - це повністю керована ("бессерверная") і нереляційна (NoSQL) служба баз даних, доступна в Amazon Web Services. Вона дозволяє надійно працювати з додатками в різних масштабах та одночасно з декількома серверами і має засоби безпеки, відновлення та резервного копіювання а також кеш пам'яті. DynamoDB справляється з піковими навантаженнями, що перевищують 20 млн запитів в секунду.[17].

DynamoDB має високу масштабованість, що означає, що ви можете почати з малого і стати дуже великим без необхідності повторного розгортання або перебудови архітектури. Він також пропонує гнучку модель, яка використовує автоматичне масштабування пропускну здатності, це означає, що він масштабує обчислювальну потужність в залежності від попиту, заощаджуючи гроші і знижуючи витрати на введення. Це робить його відмінно підходить для мобільних пристроїв, ігор, Інтернету речей та інших швидкозростаючих і об'ємних додатків.

Amazon DynamoDB пропонує безліч переваг у порівнянні з іншими системами управління базами даних NoSQL, такими як Apache Cassandra і MongoDB. Інтеграція між DynamoDB та іншими сервісами AWS особливо корисна. Якщо ви вже є користувачем AWS, це чудовий вибір.

Як бессерверная служба баз даних, настройка проста. Просто відкрийте консоль управління AWS і скористайтеся майстром. І навпаки, щоб налаштувати локальний екземпляр MongoDB, вам необхідно слідувати довгому списку інструкцій і, можливо, доведеться усунути помилки аутентифікації.

Безпека DynamoDB регулюється Управлінням ідентифікацією та доступом AWS (IAM). Ви також можете використовувати інші функції безпеки AWS для поліпшення елементів управління. Хоча MongoDB безпечний, в минулому мали місце порушення безпеки через неправильну конфігурацію та управління.

Як DynamoDB, так і MongoDB безкоштовні протягом заздалегідь визначеного періоду часу. Після закінчення терміну безкоштовного використання DynamoDB розраховує вартість на основі операцій читання і запису, в той час як MongoDB розраховує вартість відповідно до споживаним сховищем.

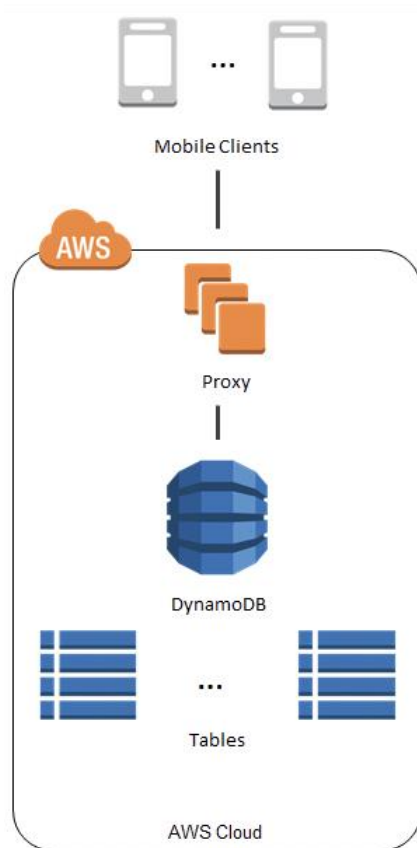


Рис.13 Приклад роботи DynamoDB.[21]

Після того як додаток проаналізувала інформацію з датчиків вона записує ці дані у базу даних DynamoDB. Ці данні можуть бути використанні у навчанні штучного інтелекту для кожного користувача особисто, що може дати більш велику вірогідність виявлення відхилень показників тіла користувача.

## 4.4 Висновки

Як бачимо була побудована інфраструктура додатку Body Sensor Network з обробкою та аналізом даних, з збереженням у базі даних та машинним навчанням.

Все це було зроблено за допомоги хмарного сервісу Amazon, а саме Amazon Kinesis, та Amazon DynamoDB. Вхідні дані були зроблені з відкритих датасетів реальних вимірів з датчиків на тілі людини. Результатом роботи є додаток Body Sensor Network, який може сповістити вас або лікарів про можливе відхилення стану здоров'я.

## 5 РОЗРОБКА СТАРТАП ПРОЕКТУ

### 1.1. Опис ідеї стартапу

Даний проект був присвячений розробці додатку Body Sensor Network за допомогою хмарних обчислювань та веб сервісу Amazon та інтеграції даних в хмарну платформу ЕСОЗ. Однак не один із наявних систем не має відкритого вихідного коду. На даному етапі розроблений додаток не може виступати як повноцінний продукт, однак, за умови подальшої розробки можна спланувати ідеї, скласти стратегію розвитку та дій для подальшого просування та створення конкурентно-спроможного продукту.

У даному розділі магістерської дисертації буде описано розробку стартапу додатку для відстеження активності, який можна використати як модуль в сторонніх рішеннях. У таблиці 5.1 показано зміст запропонованої ідеї, напрямки для застосування, переваги та вигоди у цих напрямках та різниця цієї ідеї від існуючих рішень.

## Зміст та ідея розроблюваного стартап-проекту

Зміст та ідея стартапу	Можливості та напрямки застосування	Переваги для користувача
Програма з можливістю відстеження медичного стану користувача	Використання розробленого способу у сторонніх додатках для відстеження стану пацієнта	Можливість точно відстежувати стан користувача та визначати коли стан може стати критичним
	Використання отриманих даних для виявлення відхилень та передбачення захворювань	Можливість отримати дані користувача та аналізувати їх за допомогою методів аналізу у реальному часі та за зібраними даними за проміжок часу передбачати захворювання користувача
	Надання даних для користувача та його лікаря.	Надання даних користувачу для стеженням та доглядом за його здоров'ям, лікарю для виявлення необхідних аналізів та затвердження діагнозу

Аналіз потенційних техніко-економічних переваг (тобто аналіз тих факторів, що відрізняються розроблена ідея від існуючих аналогів та замінників) показано у таблиці 5.2. Прямими аналогами є MobiHealth (у таблиці – Конкурент №1), CodeBlue (Конкурент №2).

*Таблиця 5.2.*

**Визначення слабких, сильних та нейтральних характеристик ідеї**

№	Технічні та економічні характеристики розроб. ідеї-проекту	Розроб. проект	Конкуренти		W	N	S
			1	2			
1	Можливість застосовувати у інших проектах чи платформах	+	-	-			+
2	Підтримка можливості інформувати лікаря та користувача при відхиленнях стану	+	-	-			+

Таблиця 5.2. (продовження)

№	Технічні та економічні характеристики розроб. ідей-проектів	Розроб. проект	Конкуренти		W	N	S
			1	2			
3	Підтримка можливості відстеження активності режимі з використанням GPS	+	+	+		+	
4	Візуалізація отриманих результатів	-	+	+	+		

У таблиці вище було визначено слабкі, сильні та нейтральні характеристики ідеї потенційного товару, що є фундаментальним для формування його конкурентоздатності. Як видно із таблиці, основною перевагою ідеї є можливість використання у інших проектах чи платформах, на що і варто робити ставку. Найбільшим недоліком є неможливість візуалізації отриманих даних користувача. Цей недолік необхідно врахувати у майбутньому, а саме додати можливість користувачу бачити візуалізацію їх даних за будь який проміжок часу.

## 1.2. Технологічний аудит ідеї

У таблиці 5.3 показано обрані технології для реалізації ідеї проекту. Розробка програми на мові програмування Python, з використанням фреймворка Spark та Delta-Lake для збору та аналізу даних, база даних AWS DynamoDB та AWS S3, також використання AWS Kinesis Stream для збору даних з датчиків.

## Можливість реалізації стартап проекту

№	Головна ідея для стартап проекту	Необхідні технології	Наявність	Доступність
1	Програма з можливістю відстеження фізичної активності користувача	Мова програмування Python	Є в наявності	Доступна безкоштовно
2		Фреймворки для роботи із даними Spark та Delta Lake	Є в наявності	Доступна безкоштовно
3		Алгоритм відстеження фізичної активності	Є в наявності	Доступна безкоштовно
4		База даних AWS DynamoDB, AWS S3 та AWS Kinesis Stream	Є в наявності	Доступна платно

## 1.3. Аналіз ринкових можливостей запуску стартап-проекту

У цьому пункті буде визначено ринкові можливості, які можна використати під час ринкового впровадження проекту, а також ринкові загрози, що можуть виникнути під час реалізації проекту. Цей аналіз дозволяє спланувати напрямки розвитку проекту враховуючи стан ринкового середовища, потреб клієнтів та пропозицій конкурентів. Таблиця 5.4. показує стан потенційного ринку для розроблюваного стартапу.

*Таблиця 5.4.*

#### **Характеристика існуючого ринку стартап-проекту**

<b>№</b>	<b>Показник</b>	<b>Характеристика</b>
1	Кількість головних гравців, од	> 5
2	Динаміка ринку (якісна оцінка)	У сфері додатків для стилю життя ринок знаходиться на етапі стрімкого зросту
3	Наявність обмежень для входу (вказати характер обмежень)	Велика кількість конкурентів, що пропонують більший набір можливостей для аналізу отриманих результатів
4	Специфічні вимоги до стандартизації та сертифікації	Стандартизацією та сертифікацією в сфері шифрування займається МОЗ

*Таблиця 5.5.*

### Основна характеристика потенційних клієнтів

№	Потреба у стартап-проекті	Потенційні клієнти	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Підвищений інтерес до власного здоров'я	Користувачі, що зацікавлені в покращенні свого здоров'я та самопочуття	Стан здоров'я потенційного користувача	Необхідність зручного відстеження для подальшого аналізу
2	Необхідність відстежувати та аналізувати стан свого здоров'я для подальшого використання та аналізу отриманої інформації	Клініки та лікарні яким потрібно слідкувати за станом пацієнтів, страхові компанії.	Безпосереднє використання для подовження та рятунку життя.	Можливість отримувати зібрану інформацію через використання відкритого API

*Таблиця 5.6.*

### Основні фактори загроз при виході на ринок

№	Фактор	Зміст	Реакція
1	Мала кількість функцій та можливостей	Ринок насичений подібними додатками, тому необхідно на нього вийти якомога скоріше, реалізувавши основні функції, які забезпечить конкурентоспроможний продукт	Затримка виходу на ринок для реалізації необхідного функціоналу
2	Економічний	Фінансова обмеженість, що може затримати вихід на ринок	Складання точного та чіткого бізнес-плану для залучення інвесторів

Таблиця 5.7.

**Основні фактори можливостей для розроблюваного стартап-проекту**

<b>№</b>	<b>Фактор</b>	<b>Зміст</b>	<b>Реакція</b>
1	Залучення клієнтів	Склавши чітку маркетингову кампанію, залучити клієнтів	Спланувати маркетингову кампанію, провести презентацію продукту та можливостей, що він має
2	Залучення інвесторів	Створення бізнес-плану допоможе залучити інвестиції у проект, що дозволить найняти персонал для прискорення виходу на ринок	Збільшення функціоналу, прискорений вихід на ринок

Таблиця 5.8.

**Аналіз конкуренції на ринку**

<b>Основні властивості конкурентів</b>	<b>Проявлення властивості</b>	<b>Вплив</b>
Схожість продукту з аналогами	Велике насичення продуктів на ринку з аналізом даних при невеликому апаратному стеку	Виконувати експерименти зі створенням нових функцій
Доступність до технологій, що використовуються у проекті	Доступ до технологій проекта має кожен, як до безкоштовних так і платних.	Реагувати на вихід нових продуктів на ринок, їх функціонал

Таблиця 5.9.

## Аналіз конкурентного ринку за М.Портером

	<b>Прямі конкуренти</b>	<b>Потенційні конкуренти</b>	<b>Клієнти</b>	<b>Замінники</b>
<b>Складові аналізу</b>	MobiHealh	CodeBlue	Компанії та користувачі, що зацікавлені у відстеженні та отриманні інформації про стан здоров'я	Усі додатки, що мають функціонал для аналітики даних та передбачення їх
<b>Висновки</b>	Існує велика кількість різноманітних рішень у конкурентів	Є маса додатків, що можуть виконувати такі ж операції	Постачальниками є світові лідери у сфері шифрування	Більшість клієнтів уже давно на ринку та мають налагоджені контакти з постачальниками

Таблиця 5.10.

## Основні фактори конкурентоспроможності проекту

<b>№</b>	<b>Фактор конкурентоспроможності</b>	<b>Обґрунтування</b>
----------	--------------------------------------	----------------------

1	Простий та зручний API	Враховуючи недоліки API та досвіду конкурентів, можна створити більш зручний API
2	Бета-тестування	Впровадження бета-тестування для виявлення найбільш важливих нових функцій аби максимального адаптувати продукт до потреб кінцевого споживача
3	Швидке реагування на відгуки	На етапі розробки неможливо передбачити всі нюанси застосування. Завдяки забезпеченню швидкого відгуку, можна використовувати користувачів у якості бета-тестувальників, що дозволить максимально швидко адаптувати додаток під потреби реальних клієнтів

Таблиця 5.11.

## Відносна оцінка стартап-проекту до наявних конкурентів

№	Фактор конкурентоспроможності	Бали 1-20	Оцінка у відношенні розроблюваного проекту до конкурентів						
			-3	-2	-1	0	1	2	3
1	Простий та зручний API	14		+					
2	Бета-тестування	12			+				
3	Швидке реагування на відгуки	15				+			

**SWOT-аналіз стартап-проекту**

<p>Сильні сторони:</p> <ul style="list-style-type: none"><li>- Врахування досвіду конкурентів;</li><li>- Бета-тестування для виявлення недоліків та побажань клієнтів</li></ul>	<p>Слабкі сторони:</p> <ul style="list-style-type: none"><li>- Досить велика конкуренція</li><li>- Наявність більшого функціоналу у конкурентів</li><li>- Фінансова обмеженість, що може затримати вихід на ринок</li></ul>
<p>Можливості:</p> <ul style="list-style-type: none"><li>- Отримати можливість продати ядро (алгоритм) програми</li></ul>	<p>Загрози:</p> <ul style="list-style-type: none"><li>- Відсутність конкурентного функціоналу для залучення клієнтів;</li><li>- Висока конкуренція</li></ul>

Таблиця 5.13.

## Альтернативні сценарії для впровадження розроблюваного стартап-проекту

№	Можлива альтернатива	Ймовірність отримання ресурсів	Термін реалізації
1	Відкриття вихідного коду та розповсюдження у форматі open-source	Дуже низька. Основними ресурсами будуть пожертви від відданих користувачів	Миттєво
2	Представлення проекту на конференціях та презентаціях	Висока. Представленням на конференціях можна поширити інформацію про розроблений продукт та залучити клієнтів	Від півроку до 5 років
3	Участь у благодійних організаціях, що займаються вивченням причин хвороб	Середня. Отримання ресурсів залежить у зацікавленості благодійної організації та фінансуванні цієї організації	Від місяця до кількох років

Таблиця 5.14.

**Переваги та недоліки розроблюваного стартап-проекту у порівнянні з конкурентами**

<b>№</b>	<b>Профіль цільової групи</b>	<b>Бажання та готовність сприймати продукт</b>	<b>Попит у аудиторії</b>	<b>Конкуренція на ринку</b>	<b>Складн. виходу на ринок</b>
1	Приватні компанії	Не всі готові прийняти	Необхідний вже	Середня конкуренція	Помірно-складна
2	Благодійні організації	Не всі готові прийняти одразу	Необхідний вже	Середня конкуренція	Середня
3	Дослідницькі програми	Не всі готові прийняти одразу	Необхідний вже	Середня конкуренція	Середня
<p><b>Висновок:</b> надано перевагу у роботі із приватними компаніями, однак розглядаються пропозиції від усіх цільових груп</p>					

Таблиця 5.15.

## Базова стратегія розвитку для стартап-проекту

№	Альтернатива	Обраний спосіб для розвитку та впровадження стартапу	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Просування додатку за допомогою маркетингової компанії та участі у конференціях, презентаціях, тощо	Вплив на цільову аудиторію	Безкоштовне поширення для благодійних та наукових цілей	Впровадження нового функціоналу згідно відгукам користувачів та бета-тестувальників

## Визначення базової стратегії конкурентної поведінки

№	Проект-першопрохідець	Тип розвитку	Схожість проекту з конкурентами	Стратегія поведінки на конкурентному ринку
1	Ні	Змішаний тип розвитку: пошук нових споживачів та залучення споживачів конкурентів	Продукти конкурентів схожі, тому компанія буде копіювати основні характеристики конкурентів	Оновлення функціоналу згідно побажань користувачів та впровадженню їх у конкурентів

Таблиця 5.17.

## Стратегія для позиціонування продукту

№	Вимоги до продукту від аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції стартап- проекту	Асоціації, які формують сприйняття проекту на ринку
1	Впровадження нових функцій, часті оновлення	Збільшити кількість персоналу, досліджувати поведінку користувачів конкурентів, їх побажання	Швидка реакція на недоліки, їх виправлення	Підрахунок кроків, здоровий спосіб життя, моніторинг активності

#### 1.4. Розроблення маркетингової програми стартап-проекту

Таблиця 5.18.

#### Концепція продукту

№	Потреба	Вигода	Існуюча перевага та дії для створення цієї переваги
1	Відстеження різних типів відхилень стану користувача	Використання отриманої інформації для аналізу стану здоров'я	Можливість відстежувати активність різних типів. Використовувати отриману інформацію для глибокого аналізу.
2	Детальний звіт про стан здоров'я	Можливість переглядати детальний звіт про стан здоров'я	Частина програми, що відображає ключові характеристики стану здоров'я
3	Оновлення	Врахування недоліків які виявленні та впровадження нового функціоналу	Створення служби підтримки для отримання інформації про недоліки та побажання користувачів

Таблиця 5.19.

## Три рівні моделі товару стартап-проекту

Рівень	Складові та характеристика	
<b>I. Товар за задумом</b>	Програмний алгоритм для відстеження фізичної активності, а також приклад реалізації на операційній системі Android	
<b>II. Товар у реальному виконанні</b>	Властивості/характеристики	Розмір
	1. Алгоритм для відстеження стану здоров'я	36 МБ
	2. Реалізація на хмарній платформі AWS	0 МБ
	Якість: внутрішнє автоматизоване тестування алгоритму, відправлення інформації про помилки, система зворотного зв'язку	
<b>III. Товар із підкріпленням</b>	До продажу: програмний код	
	Після продажу: запакований архів	
За рахунок чого потенційний товар буде захищено від копіювання: захист від копіювання буде забезпечено стандартними засобами та правилами розповсюдження.		

Таблиця 5.20.

## Межі ціни для стартап-проекту

<b>№ п/п</b>	<b>Рівень цін на товари-замінники</b>	<b>Рівень цін на товари-аналоги</b>	<b>Рівень доходів цільової групи споживачів</b>	<b>Верхня та нижня межі становлення ціни на товар/послугу</b>
1	20\$ - 30\$	20\$ - 30\$	> 8000\$/місяць	20,99\$-29,99\$

*Таблиця 5.21.*

### **Система для формування ринку збуту продукту**

<b>№ п/п</b>	<b>Закупівельна поведінка</b>	<b>Функції збуту постачальника</b>	<b>Глибина каналу збуту</b>	<b>Оптимальна система збуту</b>
1	Договір про надання послуг без передачі ПО до замовника	Просування та забезпечення продажу продукту, залучення нових клієнтів	Однорівневий (через роздрібну дистрибуцію)	Вертикальна (власність на продукт залишається у розробника)

Таблиця 5.22.

**Визначення маркетингових комунікацій для стартап-проектку**

№	Специфіка клієнтів	Канали зв'язку клієнтів	Позиціонування	Мета рекламного повідомл.	Концепція рекламного повідомл.
1	Купують товар на вимогу	Конференції, виступи, презентації на спеціалізованих заходах	Надання послуг у благодійних та комерційних цілях	Демонстр. можливостей додатку	Пояснення цільовій аудиторії переваг додатку та його можливостей

## ВИСНОВОК

У цьому розділі було проведено розробку стартап-проекту для програмного продукту, опис якого було представлено у попередніх розділах. У рамках цього розділу було описано ідею для стартап проекту, проведено порівняльний аналіз можливостей конкурентів та їх переваг та недоліків та вирішено наступні завдання:

1. Описано ідею та її мінімальний функціонал для виходу на ринок. Проаналізовано переваги та недоліки конкурентів, визначено переваги розробленого продукту та його можливий розвиток;
2. Проведено розгорнутий технічний аудит проекту;
3. Проведено аналіз ринку та можливості для запуску проекту на цьому ринку. Розроблено ринкову стратегію розвитку та план розвитку в умовах наявного ринку. Визначено цільові групи та способи просування проекту та продукту;
4. Описано план з маркетингового просування проекту на ринку та сформовано основний канал для продажу продукту, а також визначено цільові групи користувачів.

Отже, у цьому розділі було проведено аналіз можливостей запуску додатку у якості стартап-проекту, оцінено ризики та можливості його запуску.

## ВИСНОВКИ

В магістерській дисертації розглянуто розробку інфраструктури та додатку Body Sensor Network та інтеграції її потоків даних в E-Health

В першому розділі було описано цілі роботи і сформовані задачі, які мають бути виконані в ній. Також там була обґрунтована актуальність роботи та дослідження предметної області. Розглянута важливість питання, великі дані, прикладам використання великих даних, E-Health, Обробка даних в режимі реального часу, Хмарні технології, Натільна комп'ютерна мережа.

У другому розділі розглянуто аналіз методів та моделі інтеграції потоків даних і моделювання в режимі реального часу, що дає основне уявлення про те, як працюють аналіз даних в реальному часу та різні методи інтеграції великих потоків даних у різні системи та пристрої. Також розглянута архітектура delta lake баз даних для запису неструктурованої інформації в архітектурі додатку, розглянуто архітектуру Delta Lake бази даних для запису даних в архітектурі додатку, також розглянуто сховища які надає нам дана технологія, розглянута використання Delta Lake у різних випадках та внутрішні компоненти .

В третьому розділі проведено аналіз роботи системи ehealth та досліді взаємодії з арі хмарної платформи, розглянуто технології які використовуються для телемедицини, особливості пристроїв та систем e-health, так саме розглянуто безпека в електронній охороні здоров'я та конфіденційність. Розглянуто моделі користування E-Health, машинне навчання в електронній охороні здоров'я та особистості m-health та використання його.

В четвертому розділі була проведена реалізація додатка ban з інтеграцією потоків даних в ehealth, була побудована архітектура додатку Body Sensor Network з обробкою та аналізом даних, з збереженням у базі даних та машинним навчанням.

Таким чином був розроблений додаток для натільної комп'ютерної мережі з використанням хмарних обчислень, а саме Amazon Web Service, у режимі реального часу для ЕКОЗ. За допомогою BSN можна визначити стан та дію користувача.

Додаток має можливість передачі даних в інші платформи та системи за допомогою API та відправляти звіти по кожному користувачу.

Подальший розвиток системи передбачує:

- Вдосконалення алгоритмів пошуку відхилень стану користувача.
- Поширення області застосування додатку, наприклад у спортивній сфері або ергономіці.
- Збільшення критерій обробки та аналізу даних, для більш точних результатів.
- Розробити інтерфейс додатку для перегляду результатів.

Вдосконалити додаток алгоритмами шифрування для забезпечення безпеки конфіденційності даних.

## ПЕРЕЛІК ПОСИЛАНЬ

1. J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Computer networks, 2008. С. 2292-2330, – Дата доступу: 03.12.2021
2. TennisSense: A platform for extracting semantic information from multi-camera tennis data [Електронний ресурс]. Режим доступу: <https://ieeexplore.ieee.org/document/5201152/authors#authors> – Дата доступу: 05.12.2021
3. Big data – Режим доступу: <https://searchdatamanagement.techtarget.com/definition/big-data> – Дата доступу: 03.12.2021
4. What is cloud computing – Режим доступу <https://www.salesforce.com/products/platform/best-practices/cloud-computing/> – Дата доступу 03.12.2021
5. P. Dourish, "The parting of the ways: Divergence, data management and collaborative work," С. 230, 1995. – Дата доступу 05.12.2021
6. Wei Yi and M.B. Blake. "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities," С.72,75, 2010. – Дата доступу 04.12.2021
7. Cloud-assisted body area networks: State-of-the-art and future challenges – [Електронний ресурс]. Режим доступу: [https://www.researchgate.net/publication/271918886\\_Cloud-assisted\\_body\\_area\\_networks\\_State-of-the-art\\_and\\_future\\_challenges](https://www.researchgate.net/publication/271918886_Cloud-assisted_body_area_networks_State-of-the-art_and_future_challenges). – Дата доступу: 03.12.2021
8. M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies," 2010 С. 80-88, – Дата доступу: 04.12.2021
9. IEEE Standard for Local and Metropolitan Area Networks. Part 15.6: Wireless Body Area Networks. IEEE; New York, NY, USA: 2012. IEEE Std 802.15.6-2012.Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016). Deep Learning. MIT Press. – Дата доступу 03.12.2021
10. The Global Sensor Networks [Електронний ресурс]. Режим доступу:

<https://www.researchgate.net/publication/37427287> The Global Sensor Networks –

Дата доступа: 03.12.2021

11. Telegraphicq: Continuous dataflow processing for an uncertain world [Электронный ресурс]. Режим доступа:

<https://www.researchgate.net/publication/220988130> Telegraphicq Continuous dataflow processing for an uncertain world – Дата доступа: 03.12.2021

12. Garg M.K., Kim D.J., Turaga D.S., Prabhakaran B. Multimodal Analysis of Body Sensor Network Data Streams for Real-Time Healthcare. 2010; С. 469–478.. – Дата доступа 06.12.2021

13. Kevin A. Delin\* and Shannon P. Jackson «The Sensor Web: A New Instrument Concept» [Электронный ресурс]. Режим доступа:

<http://www.sensorwaresystems.com/nasa/resources/sensorweb-concept.pdf> – Дата доступа: 02.12.2021

14. S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, and R. Buyya, "An Autonomic Cloud Environment for Hosting ECG Data Analysis Services" 2011. [Электронный ресурс]. Режим доступа: <http://www.cloudbus.org/papers/ECGCloudApp.pdf> – Дата доступа: 07.12.2021

15. BodyCloud: A SaaS Approach for Community Body Sensor Networks [Электронный ресурс]. Режим доступа: <https://fddocuments.in/document/bodycloud-a-saas-approach-for-community-body-sensor-networks.html>– Дата доступа: 05.12.2021

16. Amazon Kinesis [Электронный ресурс]. Режим доступа: [https://aws.amazon.com/ru/kinesis/?nc2=h\\_ql\\_prod\\_an\\_kin](https://aws.amazon.com/ru/kinesis/?nc2=h_ql_prod_an_kin) – Дата доступа 03.12.2021

17. Amazon DynamoDB– Режим доступа: [https://aws.amazon.com/ru/dynamodb/?nc2=h\\_ql\\_prod\\_db\\_ddb](https://aws.amazon.com/ru/dynamodb/?nc2=h_ql_prod_db_ddb)– Дата доступа 03.12.2021

18. Wireless Body Sensor Networks: A Review – [Электронный ресурс]. Режим доступа:

[https://www.researchgate.net/publication/283739667\\_Wireless\\_Body\\_Sensor\\_Networks\\_A\\_Review](https://www.researchgate.net/publication/283739667_Wireless_Body_Sensor_Networks_A_Review) – Дата доступа: 06.12.2021

19. Programming Wireless Body Sensor Network Applications through Agents. – [Электронный ресурс]. Режим доступа:

[https://www.researchgate.net/publication/220866195\\_Programming\\_Wireless\\_Body\\_Sensor\\_Network\\_Applications\\_through\\_Agents](https://www.researchgate.net/publication/220866195_Programming_Wireless_Body_Sensor_Network_Applications_through_Agents) – Дата доступа: 06.12.2021

20. A Survey of Body Sensor Networks – Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690007/> – Дата доступа 03.12.2021

21. Amazon . – Режим доступа: [https://aws.amazon.com/ru/?nc2=h\\_lg](https://aws.amazon.com/ru/?nc2=h_lg) – Дата доступа 04.12.2021

22. The applications of eHealth technologies in the management of asthma and allergic diseases – Режим доступа : <https://onlinelibrary.wiley.com/doi/10.1002/ctt2.12061> - Дата доступа 02.12.2021

23. The Benefits and Challenges of E-Health Applications in Developing Nations: A Review. – Режим доступа: [https://www.researchgate.net/publication/329877708\\_The\\_Benefits\\_and\\_Challenges\\_of\\_E-Health\\_Applications\\_in\\_Developing\\_Nations\\_A\\_Review](https://www.researchgate.net/publication/329877708_The_Benefits_and_Challenges_of_E-Health_Applications_in_Developing_Nations_A_Review) – Дата доступа 02.12.2021

24. Using Artificial Intelligence Technology to Solve the Electronic Health Service by Processing the Online Case Information. – Режим доступа: [https://www.researchgate.net/publication/356576633\\_Using\\_Artificial\\_Intelligence\\_Technology\\_to\\_Solve\\_the\\_Electronic\\_Health\\_Service\\_by\\_Processing\\_the\\_Online\\_Case\\_Information](https://www.researchgate.net/publication/356576633_Using_Artificial_Intelligence_Technology_to_Solve_the_Electronic_Health_Service_by_Processing_the_Online_Case_Information) – Дата доступа 03.12.2021

25. The Fundamentals of Data Warehouse + Data Lake = Lake House. – Режим доступа: <https://towardsdatascience.com/the-fundamentals-of-data-warehouse-data-lake-lake-house-ff640851c832> – Дата доступа 04.12.2021

26. Open Sourcing Delta Lake. – Режим доступа: <https://databricks.com/blog/2019/04/24/open-sourcing-delta-lake.html> – Дата доступа 05.12.2021

27. Аналіз даних і процесів: навч. посібник. 3-е. – Режим доступу:  
<http://kist.ntu.edu.ua/textPhD/AnalizDanyhIProcessov.pdf> – Дата доступу 01.12.2021