**Відслідковування потоку фроду на мережі мобільного оператора**

Підготували

студенти групи СМПР-1:

Дар’я Плющ

Денис Півень

Андрій Ляшкевич

**Зміст**

[1. Постановка задачі 3](#_Toc121468290)

[2. Підхід до рішення 4](#_Toc121468291)

[3. Опис рішення 5](#_Toc121468292)

[3.1. Моделювання та стрімінг даних 5](#_Toc121468293)

[3.2. База даних 5](#_Toc121468294)

[3.3. Валідація та обробка даних 6](#_Toc121468295)

[3.4. Аналітика і дашборд 6](#_Toc121468296)

1. Постановка задачі

Фрод — розсилка фішингових СМС, які спонукають терміново перевести гроші чи видати секретні дані доступу до рахунку. У фішингових розсилках шахраї, в основному, використовують підроблені посилання, замасковані під справжні, а саме повідомлення грає на емоціях і складається таким чином, щоб викликати паніку. Наприклад, мова може йти про відновлення доступу до банківського рахунку.

Інший варіант фроду — розсилка СМС з посиланнями на вірусні сайти, що крадуть гроші з особистого рахунку. Приводом для переходу на невідомий сайт може послужити, наприклад, повідомлення про виграш у лотереї.

При виконанні лабораторної перед нами стояли наступні завдання:

1. Створення і запуск стрімінгу запитів в мережі мобільного оператора;
2. Створення і налаштування бази даних для збереження інформації про запити;
3. Підключення до потоку, отримання з нього об’єктів, їх валідація, аналіз та запис в базу даних;
4. Аналітика даних та створення дашборду.
5. Підхід до рішення

Очевидним є те, що жоден мобільний оператор не надасть дані своїх абонентів, тому ми власноруч моделювали дані які надсилались потоком.

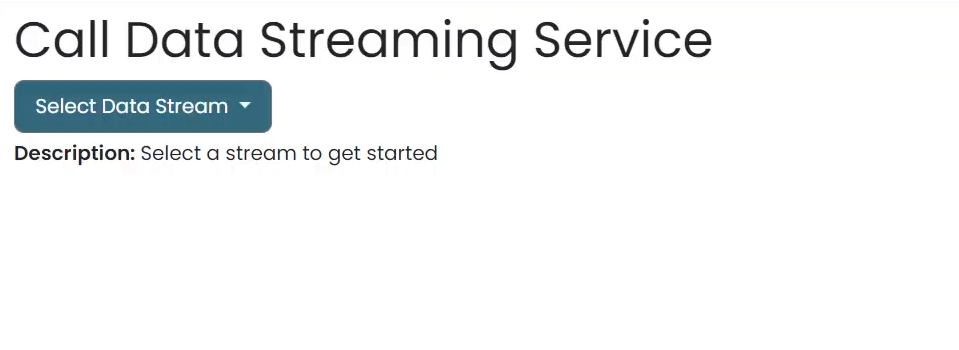
Для реалізації лабораторної ми обрали наступні технології та сервіси:

* Мова програмування – JavaScript
* Стрімінг – PubNub
* База даних – MongoDB
* Дашборди і аналітика – Tableau

GitHub репозиторій: <https://github.com/PivDen2000/Modern-Programming-Technologies-LAB>

1. Опис рішення
   1. Моделювання та стрімінг даних

Стрімінг даних реалізовано за допомогою сервісу PubNub у вигляді веб-сторінки наступного вигляду:

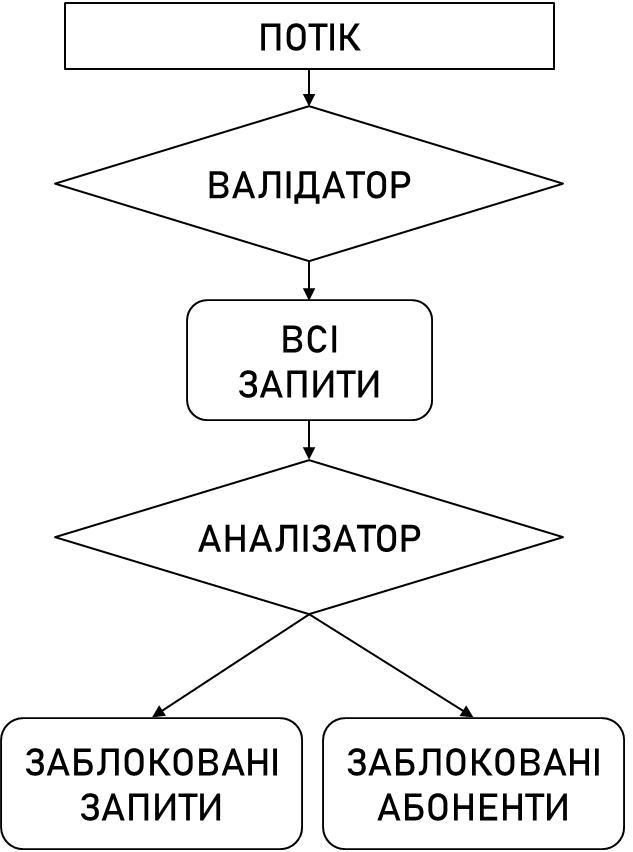


В потоці надсилаються об’єкти JSON з інформацією про мобільний запит:

* номер абонента що надсилає запит
* номер отримувача запиту
* тариф абонента що надсилає запит
* тип запиту – дзвінок чи SMS-повідомлення
* дата і час запиту
  1. База даних

В MongoDB створено базу із трьома таблицями для збереження наступної інформації:

* таблиця всіх запитів в якій зберігаються всі об’єкти що ми отримуємо з потоку, тобто:
  + номер абонента що здійснює запит;
  + номер абонента до якого здійснюється запит;
  + тариф абонента що здійснює запит;
  + тип запиту – дзвінок чи SMS-повідомлення ;
  + дата і час запиту;
* таблиця заблокованих запитів в яку зберігаємо тільки інформацію про підозрілі запити;
* таблиця із потенційними фродерами в якій зберігається інформація про них:
  + номер телефону;
  + тариф абонента;
  + тип запиту, через який заблокували даного користувача;
  + час і дата блокування.
  1. Валідація та обробка даних

Після отримання JSON об’єктів із потоку відбувається базова валідація отриманих об’єктів. В кожному об’єкті проходить перевірка номерів чи є вони цілими числами із множини [380000000000; 380999999999].

Всі об’єкти що пройшли валідацію записуються в таблицю із всіма запитами, інші ж вважаються недійсними чи помилковими і з ними нічого не відбувається.

Теоретично таким чином можуть втрачатись певні запити які мають блокуватись і завдяки яким будуть виявлятись фродери, але якщо враховувати величину мережі мобільного оператора і те що імовірність отримування пошкоджених даних дуже мала, можемо знехтувати цією похибкою.

Після запису об’єкта в таблицю всіх запитів кожен з цих об’єктів аналізується на підозру у фродінгу.

Якщо запит задовольняє всі умови підозрілості, то він записується в таблицю заблокованих запитів, а дані абонента що здійснив цей запит записуються в таблицю заблокованих абонентів.

* 1. Аналітика і дашборд

З використанням програмного забезпечення Tableau було створено дашборди, в яких відображається аналітика за різними даними з фільтрами. Вони мають наступний вигляд:

