

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Дослідження та розробка програмного модуля аналізу  
та моніторингу активності користувачів»

на здобуття освітнього ступеня магістра  
зі спеціальності 122 Комп'ютерні науки  
(код, найменування спеціальності)  
освітньо-професійної програми Комп'ютерні науки  
(назва)

*Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ Роман ГОРКУН  
(підпис) (Ім'я, ПРІЗВИЩЕ здобувача)

Виконав:  
здобувач вищої освіти  
група КНДМ-61

Роман ГОРКУН

Керівник:  
науковий ступінь,  
вчене звання

Сергій СЕРИХ  
доц, к.т.н

Рецензент:  
науковий ступінь,  
вчене звання

\_\_\_\_\_  
(Ім'я, ПРІЗВИЩЕ)

**Київ 2023**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Комп'ютерних наук

Ступінь вищої освіти Магістр

Спеціальність Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедру Комп'ютерних наук

\_\_\_\_\_ Віктор ВИШНІВСЬКИЙ

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Горкуну Роману Дмитровичу

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Дослідження та розробка програмного модуля аналізу та моніторингу активності користувачів

керівник кваліфікаційної роботи Сергій СЕРИХ, доц., к.т.н

*(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, методи моніторингу та аналізу діяльності користувачів, вимоги до систем UBA, машинне навчання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження сучасних систем аналізу та моніторингу активності користувачів

Постановка задачі та вибір технічних засобів реалізації

Програмна реалізація серверної частини системи UBA

5. Перелік графічного матеріалу: *презентація*

1. Призначення та основні функціональні можливості систем UBA
2. Типи аналітики користувача
3. Специфікація вимог
4. Діаграми прецедентів
5. Структура серверної частини системи UBA
6. Ефективна обробка зібраних даних
7. Аналіз даних активності користувачів
8. Оптимізація стратегії електронного маркетингу за допомогою машинного навчання
9. Висновки

6. Дата видачі завдання «19» жовтня 2023 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	19.10-05.11.23	
2	Дослідження сучасних методів аналізу та моніторингу діяльності користувачів	05.11-12.11.23	
3	Встановлення відмінних особливостей, яких не вистачає існуючим системам	13.11-19.11.23	
4	Розробка діаграми прецедентів та проектування структури системи	20.11-25.11.23	
5	Розроблення модулю збору даних	27.11-03.12.23	
6	Розроблення аналітичного модулю	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Роман ГОРКУН

(Ім'я, ПРІЗВИЩЕ)

Керівник  
кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Сергій СЕРИХ

(Ім'я, ПРІЗВИЩЕ)





## РЕФЕРАТ

Текстова частина бакалаврської роботи: 69 с., 1 табл., 16 рис., 20 джерел.

*Наукове завдання* – розробка серверної частини універсального модуля аналізу та моніторингу поведінки користувачів.

*Мета роботи* – окреслити шляхи підвищення ефективності аналізу та збору даних про активність користувачів.

*Об'єкт дослідження* – процес аналізу та моніторингу активності користувачів.

*Предмет дослідження* – система аналізу та моніторингу активності користувачів.

*Короткий зміст роботи:* Проведено аналіз призначення, встановлено вимоги до функціональності, розглянуто доступні методи аналізу та моніторингу, які використовуються в схожих системах, порівняно існуючі системи.

Визначена недостатня ефективність та універсальність сучасного рівня аналізу та моніторингу активності користувачів, сформульовано вимоги до модуля, підібрані підходящі технології та обґрунтовано технічний стек.

Створені діаграми прецедентів, спроектовано підходящу структур, вибрані кращі способи збору даних та методи аналізу.

Розроблена система є кросплатформеною, забезпечує швидку пропускну здатність обробки великих об'ємів даних, надає можливість легко масштабуватись, має основні функції необхідні для кожного проекту з потребою аналізу та моніторингу, дозволяє скоротити час та сконцентруватися на налаштуванні та кастомізації під індивідуальну бізнес-логіку.

**КЛЮЧОВІ СЛОВА:** МОНІТОРИНГ, ЗБІР-ДАНИХ, ПОВЕДІНКА КОРИСТУВАЧА, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ, ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ.

## ABSTRACT

Text part of the master's qualification work: 69 pages, 16 pictures, 1 table, 20 sources.

*The research task* – to develop the server side of a universal module for analysing and monitoring user behaviour.

*The purpose of the work* – to outline ways to improve the efficiency of analysis and collection of data on user activity.

*Object of research* – the process of analysing and monitoring user activity.

*The subject of research* – system for analysing and monitoring user activity.

*Summary of work:* the purpose of the work was analysed, the requirements for functionality were established, the available methods of analysis and monitoring used in similar systems were considered, and the existing systems were compared.

The lack of efficiency and universality of the current level of analysis and monitoring of user activity was determined, requirements for the module were formulated, suitable technologies were selected and the technical stack was substantiated.

Precedent diagrams were created, a suitable structure was designed, and the best methods of data collection and analysis methods were selected.

The developed system is cross-platform, provides fast throughput for processing large amounts of data, provides the ability to easily scale, has the basic functions necessary for each project requiring analysis and monitoring, allows to reduce time and concentrate on setting up and customizing for individual business logic.

**KEYWORDS:** MONITORING, DATA COLLECTION, USER BEHAVIOUR, INTELLIGENT ANALYSIS, PERSONALISED RECOMMENDATIONS.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	10
ВСТУП.....	11
1 ДОСЛІДЖЕННЯ СУЧАСНИХ СИСТЕМ АНАЛІЗУ ТА МОНІТОРИНГУ АКТИВНОСТІ КОРИСТУВАЧІВ .....	12
1.1 Призначення та основні функціональні можливості систем UBA.....	12
1.2 Методи моніторингу діяльності користувача.....	14
1.3 Методи аналізу діяльності користувача.....	16
1.3.1 Дескриптивна аналітика .....	16
1.3.2 Сегментна аналітика .....	17
1.3.3 Аналітика воронки .....	18
1.3.4 Когортна аналітика.....	19
1.4 Аналіз та порівняльна характеристика існуючих систем UBA .....	20
1.4.1 Google Analytics.....	20
1.4.2 Hotjar .....	22
1.4.3 Adobe Analytics .....	24
1.4.4 Індивідуальна система UBA.....	27
2 ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР ТЕХНІЧНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ .....	29
2.1 Постановка задачі щодо розробки універсального модуля UBA.....	29
2.2 Вибір підходящих технологій для розробки універсального модуля UBA... ..	30
2.2.1 Мова програмування Java.....	30
2.2.2 Фреймворк Spring Boot.....	32
2.2.3 База даних MongoDB .....	33
2.2.4 Брокер повідомлень Apache Kafka .....	34
2.2.5 Бібліотека Weka.....	36
2.2.6 Збірник проєктів Gradle .....	37
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ СИСТЕМИ UBA .....	40
3.1 Розробка діаграми прецедентів .....	40
3.2 Проектування структури системи UBA .....	44



3.3 Розробка модуля збору даних.....	48
3.3.1 Збір даних через HTTP-запити.....	48
3.3.2 Збір даних через брокер повідомлень .....	53
3.4 Розробка аналітичного модуля.....	55
3.4.1 Статистичні методи аналізу .....	56
3.4.2 Методи машинного навчання .....	59
ВИСНОВКИ.....	67
ПЕРЕЛІК ПОСИЛАНЬ .....	68
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....	70

## ПЕРЕЛІК СКОРОЧЕНЬ

UBA - User Behavior Analytics - аналітика поведінки користувачів.

JVM - Java Virtual Machine – віртуальна машина Java.

HTTP - HyperText Transfer Protocol – мережевий протокол передачі даних прикладного рівня.

HTTPS - Hypertext Transfer Protocol Secure - розширення протоколу HTTP для підтримки шифрування.

JSON - JavaScript Object Notation – текстовий формат передачі даних, заснований на JavaScript.

BSON - Binary JavaScript Object Notation - формат обміну даними, двійкова форма представлення простих структур даних і асоціативних масивів.

REST - Representational State Transfer - архітектурний стиль програмного забезпечення для розподілених систем.

URI - Uniform Resource Identifier - рядок символів, який використовується для ідентифікації будь-якого ресурсу на його адресу.

API - Application Programming Interface – механізм, який дозволяє програмним компонентам взаємодіяти між собою.

ML - Machine Learning - збір методів в області штучного інтелекту для створення машини, яка вчиться на власному досвіді.

## ВСТУП

У сучасному світі, де інтернет вже став його невід'ємною частиною, важливість аналізу та моніторингу даних про активність користувачів набуває все більшого значення. Поширення цифрових технологій породжують нові виклики та можливості для дослідження поведінки користувачів, щоб в подальшому використовувати цю інформацію у різних сферах, включаючи інтернет-маркетинг, покращення клієнтського досвіду та розвиток веб-ресурсів та додатків.

З точки зору бізнесу, правильно спроектований та оптимізований продукт, адаптований до потреб користувачів, забезпечує збільшення конверсії, збільшення продажів та залучення нових клієнтів. Підприємства, які вміють аналізувати та реагувати на активність користувачів, отримують конкурентну перевагу та можуть швидше підлаштовуватись під зміни в ринкових умовах.

Автори контенту, такі як блогери, можуть використовувати різні системи аналізу для більш ефективної взаємодії зі своєю аудиторією та підвищення якості свого контенту. Використання аналітичних інструментів виявлення тенденцій у поведінці аудиторії може допомогти у визначенні того, які види контенту найбільш успішні і популярні, і дасть можливість на основі цього зворотного зв'язку покращувати контент у майбутньому.

Отже, використання програмного модуля для аналізу та моніторингу активності користувачів стає стратегічно важливою для будь-якої організації, яка прагне не лише існувати в цифровому світі, але й процвітати та розвиватися.

# 1 ДОСЛІДЖЕННЯ СУЧАСНИХ СИСТЕМ АНАЛІЗУ ТА МОНІТОРИНГУ АКТИВНОСТІ КОРИСТУВАЧІВ

## 1.1 Призначення та основні функціональні можливості систем UBA

Аналіз та моніторинг діяльності користувача - це процес збору, обробки та аналізу даних про взаємодію користувачів з веб-системами, додатками, онлайн-платформами або іншими інформаційними системами з метою отримання інформації про їхню поведінку, зрозуміння їхніх потреб, вподобань та взаємодій з певними елементами програмного продукту.

Це дозволяє організаціям отримати важливі відомості щодо попередніх взаємодій із клієнтами, допомагає вдосконалювати продукти або послуги, надаючи більш якісний та зручний досвід для користувачів, а також використовується для прийняття більш ефективних рішень в галузі маркетингу та продажів.

Призначення моніторингу діяльності користувача:

- Поліпшення користувацького досвіду: моніторинг дозволяє компаніям виявляти проблеми та недоліки в інтерфейсі, навігації та функціональності своїх додатків. Це надає можливість вчасно реагувати на них, виправляти помилки та покращувати продукт, що врешті-решт покращує досвід користувачів і підвищує їхню лояльність.
- Підвищення ефективності роботи систем: аналіз користувацької поведінки виявляє закономірності та тенденції взаємодії з продуктом. Ця інформація може бути використана для оптимізації і покращення існуючих функцій та розробки нових, що відповідають потребам кінцевих користувачів.
- Прийняття рішень у галузі маркетингу та продажів: моніторинг збирає дані про користувацький попит та переваги. Це допомагає в розробці більш цілеспрямованих маркетингових кампаній та стратегій продажу, що підвищує конверсію та прибутковість компанії, наприклад дає змогу

компаніям створювати персоналізовані пропозиції та рекомендації для кожного користувача, що підвищує рівень лояльності клієнтів.

- Захист від шахрайства та запобігання фінансовим збиткам: відстеження допомагає виявляти незвичайну чи підозрілу активність користувачів, що може свідчити про можливі шахрайські дії. Вчасне виявлення цих ситуацій дозволяє компаніям приймати заходи для запобігання фінансовим збиткам та забезпеченню безпеки клієнтів.

Системи аналізу активності користувачів надають комплексний інструментарій для збирання, обробки та інтерпретації даних, пов'язаних із взаємодією користувачів з різними платформами. Розглянемо основні функціональні можливості, якими може володіти такий сервіс:

1. Збір та агрегація даних. Система збирає велику кількість даних щодо активності користувача, таких як перегляди сторінок, кліки, час перебування на сторінці, та інші взаємодії.
2. Аналіз поведінки користувачів. Модуль використовує аналітичні методи для виявлення закономірностей у поведінці користувачів, таких як популярні сторінки, шляхи навігації, та інші показники, що допоможуть зрозуміти користувацькі потреби та інтереси.
3. Сегментація користувачів. Система може класифікувати користувачів за різними критеріями, такими як географічне розташування, вік, інтереси тощо. Це дозволяє визначити специфічні групи користувачів і персоналізувати взаємодію з ними.
4. Реагування на зміни. Системи моніторингу можуть автоматично виявляти зміни у поведінці та сповіщати про них. Це дозволяє організаціям вчасно реагувати на зміни у вимогах ринку та користувацьких уподобаннях.
5. Персоналізовані рекомендації. На основі аналізу даних модуль може забезпечувати можливість надання користувачам персоналізованих рекомендацій, зокрема товарів, послуг чи контенту, що відповідає їхнім інтересам.

6. Візуалізація даних. Система повинна надавати інтерактивні інтерфейси для візуалізації даних у вигляді графіків, діаграм, звітів тощо, щоб зробити аналіз користувацької активності більш доступним та зрозумілим для користувачів.
7. Системи оповіщень. Модуль може включати систему оповіщень для адміністраторів про незвичайні або важливі активності користувачів, дозволяючи вчасно реагувати на можливі проблеми.
8. Захист даних і конфіденційність. Модуль повинен забезпечувати безпеку зібраних даних та дотримання принципів конфіденційності користувачів, відповідаючи стандартам і правилам захисту особистих даних.
9. Можливість розширення та індивідуальна конфігурація. Модуль повинен бути готовим до розширення функціональності в майбутньому та надавати можливість налаштування для персональних потреб різних веб-сайтів чи програмних продуктів. Система може бути інтегрована з іншими інформаційними системами компанії для обміну даними та взаємодії з ними, наприклад, системами управління відносинами з клієнтами, системами управління контентом тощо.

## **1.2 Методи моніторингу діяльності користувача**

Активність користувачів в інформаційних системах являє собою невичерпне джерело інформації для виявлення вкрай корисної інформації для бізнесу. Моніторинг є комплексним підходом, що містить різноманітні способи та інструменти для відстеження дій користувачів у системах. Розглянемо базові методи відстеження діяльності споживача.

Логування подій – процес запису подій, що відбуваються в інформаційній системі або програмному забезпеченні, до спеціальних файлів або баз даних, які називаються логами. Цей метод моніторингу широко застосовується з метою забезпечення безпеки, аналізу продуктивності та виявлення проблем.

Моніторинг активності мережі – процес відстеження та аналізу мережевих операцій, які виконуються користувачами або пристроями в комп'ютерній мережі. Цей метод дозволяє організаціям та системним адміністраторам стежити за тим, які ресурси використовуються, хто і як взаємодіє в мережі, та виявляти потенційні загрози безпеці. Використовуючи цей спосіб, адміністратор може ідентифікувати потоки даних між пристроями, виявляти пікові навантаження, запобігати атакам відмови в обслуговуванні та інші види мережних загроз.

Моніторинг активності веб-сайту – процес збору, аналізу та відстеження даних про взаємодію з веб-ресурсом. Цей метод спрямований на розуміння поведінки відвідувача сайту, його вподобань, а також на виявлення тенденцій, які можуть бути корисні для поліпшення досвіду користувача та ефективності веб-ресурсу. Приклади інформації, яку можна зібрати в такий спосіб: час відвідування, шляхи навігації сайтом, ключові точки взаємодії, перегляди сторінок та час, проведений на кожній сторінці.

Моніторинг операційної системи – процес контролю діяльності співробітників в інформаційному середовищі. Таким чином можна зібрати дані про активність у різних додатках, частоту використання певних функцій, історію навігації у файловій системі, а також дані про ресурси системи, такі як використання процесора, оперативної пам'яті та дискового простору.

Опитування та зворотний зв'язок – процес збирання інформації безпосередньо від користувачів через різні форми зворотного зв'язку. Це дозволяє зрозуміти емоційний настрій, отримати прямі коментарі та оцінки користувачів щодо різних аспектів продукту чи послуги, виявити конкретні побажання.

Однак збір кожної точки даних не завжди є ефективним. Залежно від вашої стратегії моніторингу активності користувачів, набагато ефективніше зосередитися на конкретних точках даних, ніж використовувати універсальний підхід або губитися в нескінченному об'ємі даних.

## 1.3 Методи аналізу діяльності користувача

Аналітика даних — це процес вивчення та обробки даних з метою отримання корисної інформації, формування висновків та підтримки прийняття рішень. Збільшення обсягів цифрової інформації наголошують на важливості ефективного аналізу. Скільки б даних ви не зібрали, без аналітики вони марні.

Для ухвалення рішень на основі даних необхідно чітко бачити всі аспекти свого продукту. Аналітичні методи дозволяють систематизувати, інтерпретувати та використовувати дані, зібрані в результаті користувальницької активності. Кожен із цих методів робить свій внесок у розуміння патернів поведінки користувачів та надає цінну інформацію.

Існує наступні типи аналітики користувача:

- Дескриптивна аналітика
- Сегментна аналітика
- Аналітика воронки
- Когортна аналітика

### 1.3.1 Дескриптивна аналітика

Дескриптивний аналіз даних — це метод аналізу даних, який використовується для опису даних. Він не передбачає жодних висновків чи передбачень про дані, а просто надає інформацію про їх характер та структуру.

Дескриптивна аналітика є фундаментальним інструментом для розуміння та опису ключових аспектів взаємодії користувачів із цифровими платформами. Її застосування надає детальний опис того, як користувачі взаємодіють із продуктом або послугою.

Аналіз починається із огляду загальної активності. Це включає виявлення основних метрик, таких як загальна кількість користувачів, кількість сеансів, динаміку зростання і падіння активності. Ці дані надають високорівневе уявлення про генеральну картину взаємодії клієнтів.



Це допомагає зрозуміти, які конкретні елементи чи послуги залучають користувачів найбільше. Наприклад, можна визначити, які функції програми використовуються частіше, а які залишаються незатребуваними. Ця інформація є цінною для оптимізації продукту та фокусування розробки на найбільш цінних функціях.

### **1.3.2 Сегментна аналітика**

Сегментація клієнтів – це процес поділу користувачів на групи на основі їх загальної поведінки для забезпечення кращого досвіду.

Отже, сегментна аналітика передбачає аналіз користувачів кожного сегмента споживачів. Ви можете визначити сегменти за користувачами, даними опитувань, подіями використання та навіть зовнішніми джерелами, такими як дані із систем управління продажами та маркетингом, реферальних каналів.

Більшість інструментів аналітики користувача дозволяють сегментувати користувачів на основі наступних моделей:

- Демографічні показники – включають стать, дохід та вік. Докладніші схеми включають національність, рід занять, расу і навіть спосіб життя.
- Поведінка в програмі – показує, чи активували користувачі програму, чи завершили вони певний UX-потік, чи виконали якісь події користувача або взаємодіяли з новою функцією.
- Обліковий запис – розділяє клієнтів на нових, корпоративних та досвідчених користувачів.
- Атрибути користувача – включають тарифний план, на якому працює користувач, якій мові він надає перевагу, які пристрої використовує, чи зареєструвався він через члена компанії або самостійно.
- Настрій користувачів – використовуючи метрику Net Promoter Score, ця модель поділяє користувачів на промоутерів (лояльних користувачів та прихильників), пасивів (байдужих користувачів) та відщепенців (невдоволених користувачів).

Компанія може використовувати дані щодо поведінки користувачів, отримані в результаті аналізу сегментів, для надання індивідуальних рішень для кожного сегменту. Крім того, можна виявити сегменти, що знаходяться на межі відтоку, та вжити превентивних заходів.

### 1.3.3 Аналітика воронки

Воронка (також відома як воронка продажів або воронка конверсії) – це ряд кроків, що використовуються для складання схеми руху користувачів до конверсії. Це може бути будь-який тип конверсії, наприклад, покупка чи реєстрація. На рисунку 1.1 показана спрощена схема воронки продажів.



Рисунок 1.1 – Схема воронки продажів

Аналітика воронки – це процес визначення користувачів, які конвертуються на кожному етапі воронки. Це дозволяє приймати обґрунтовані рішення для поліпшення досвіду користувача і, відповідно, коефіцієнта конверсії. Ви можете використовувати свої сильні сторони, виділивши області з висококонверсійним трафіком.

Більше того, аналітика воронки дозволяє проводити кампанії повторного залучення, змінювати потоки онбордингу або виправляти помилки в будь-якій точці потоку користувачів.

Також, дана аналітика виходить за рамки покращення показників конверсії та допомагає зрозуміти, як вони різняться залежно від поведінки користувача. Таким чином, ви можете дізнатися, що стимулює конверсію, що змушує користувачів відторгатися до конверсії і оптимізує ефективність воронки.

### **1.3.4 Когортна аналітика**

Когортна аналітика дозволяє побачити, що когорта (підсекція користувачів) робить у вашому продукті. Компанії-розробники програмного забезпечення зазвичай використовують когортну аналітику для виміру відтоку клієнтів. Саме тому когортний аналіз називають аналізом відтоку клієнтів.

Існує 2 найбільш поширені види когортної аналітики:

- Когорта придбання: коли сегмент клієнтів створюється на основі дати реєстрації.
- Поведінкова когорта: коли когорта ґрунтується на поведінці користувачів під час використання ними вашого продукту.

Залежно від вибраного вами типу, когортна аналітика допоможе вам відповісти на такі запитання:

- Який сегмент користувачів найчастіше відторгається?
- Як довго ви можете утримувати середньостатистичного клієнта і коли він залишає ваш продукт?

Для проведення когортного аналізу вам знадобиться таблиця когорт, подібна до тієї, що показана на рисунку 1.2.

Рядки (згори донизу) сегментують когорти на основі дат їх реєстрації, зокрема, по місяцях. Колонки (зліва направо) показують, скільки часу минуло з підписки клієнта. У кожному осередку таблиці вказано відсоток клієнтів, які відгукнулися протягом місяця після оформлення передплати.

Знаючи, коли користувачі йдуть і як змінюється коефіцієнт відтоку залежно від того, коли реєструється нова когорта, ви можете перевірити, наскільки ефективні ваші маркетингові та продуктові стратегії.

Percentage of Customers Retained Relative to Previous Month													
	Cohort Value	0	1	2	3	4	5	6	7	8	9	10	11
Apr 2022	21	100%	100%	95.24%	95%	100%	100%	100%	100%	94.74%	100%	100%	100%
May 2022	30	100%	100%	100%	93.33%	100%	100%	100%	96.43%	96.3%	100%	103.85%	
Jun 2022	30	100%	100%	100%	90%	100%	96.3%	100%	100%	96.15%	104%		
Jul 2022	28	100%	100%	100%	92.86%	100%	100%	100%	100%	100%			
Aug 2022	35	100%	97.14%	100%	88.24%	103.33%	100%	96.77%	96.67%				
Sep 2022	53	100%	100%	100%	92.45%	100%	100%	97.96%					
Oct 2022	44	100%	100%	97.73%	93.02%	95%	97.37%						
Nov 2022	48	97.92%	100%	93.62%	90.91%	92.5%							
Dec 2022	49	100%	93.88%	97.83%	91.11%								
Jan 2023	67	100%	100%	100%									
Feb 2023	62	100%	100%										
Mar 2023	80	100%											
Average	46	99.82%	99.14%	98.5%	91.74%	98.46%	99.08%	98.88%	98.45%	96.94%	101.45%	102.27%	100%

Рисунок 1.2 – Таблиця когорт

Крім того, ви зможете побачити, чи впливають зміни, які ви внесли в процес прийому нових користувачів, на коефіцієнт утримання користувачів в перший місяць. Ви навіть можете побачити, чи надають зміни у вашому продукті, UX або маркетинговій стратегії будь-який вплив, позитивний чи негативний, на ваш продукт.

## 1.4 Аналіз та порівняльна характеристика існуючих систем UBA

### 1.4.1 Google Analytics

Google Analytics – це потужний інструмент для створення детальної статистики відвідувачів веб-сайтів. Він надає детальну інформацію про те, як користувачі взаємодіють з веб-сайтом, що дозволяє власникам сайтів і маркетологам приймати більш обдумані рішення для покращення досвіду користувача та ефективності маркетингових кампаній.

Google Analytics збирає дані про відвідувачів веб-сайту або програми за допомогою JavaScript-коду, який розміщується на сторінках сайту або програми. Основні типи даних, які збирає цей інструмент:

- Інформація про відвідувачів: унікальні відвідувачі та ті, що повертаються, географічний розподіл відвідувачів, технічна інформація про браузери та пристрої.
- Джерела трафіку, звідки прийшли відвідувачі (наприклад, пошукові системи, прямий трафік, посилання з інших сайтів, рекламні кампанії, соціальні мережі).
- Поведінка на сайті: сторінки, що переглядаються, час перебування на кожній, взаємодії з контентом (наприклад, перегляд відео, кліки за посиланнями), показники відмов (відвідувачі, що залишили сайт після перегляду тільки однієї сторінки).
- Конверсії та цілі: виконання цілей, визначених власником сайту (наприклад, заповнення форми, здійснення покупки, підписка на розсилку).
- Дані електронної комерції: інформація про продаж, включаючи кількість проданих товарів, виручку, середній чек та інші метрики електронної комерції.
- Мобільний трафік: інформація про те, як користувачі взаємодіють із сайтом через мобільні пристрої, включаючи типи пристроїв та операційні системи.

Google Analytics використовує різні методи аналізу даних, щоб допомогти власникам сайтів та маркетологам розуміти поведінку користувачів та оптимізувати свої стратегії маркетингу та веб-розробки. Основні методи аналізу, які надаються: сегментація даних, аналіз воронки конверсій, аналіз багатовимірних даних, аналіз поведінки, аналіз електронної комерції.

### Недоліки Google Analytics:

1. Можливість підробки даних. Код відстеження доступний публічно, тому можуть виникнути спроби підробки даних. Наприклад, спам-боти можуть створювати помилкові сесії, що може спотворювати аналітику.
2. Обмеження семплювання. У випадку великих обсягів даних, Google Analytics може використовувати семплювання, що означає, що аналіз проводиться на основі випадкової вибірки даних, а не повного обсягу. Це може знижувати точність результатів.
3. Залежність від JavaScript. Для збору даних інструмент потребує підтримки JavaScript у браузері користувача. Якщо користувач вимкне JavaScript, це може призвести до неповного збору даних.
4. Обмежені можливості мобільних додатків. Система добре інтегрована з веб-сайтами, але має обмежені можливості для аналізу мобільних додатків у порівнянні з деякими спеціалізованими аналітичними інструментами.
5. Не всі види взаємодії відстежуються. Деякі складні види взаємодії (наприклад, взаємодія з елементами сторінки, які не є посиланнями) можуть вимагати використання додаткових інструментів.

### 1.4.2 Hotjar

Hotjar – це онлайн-сервіс для аналітики сайту та накопичення зворотного зв'язку, який може допомогти покращити юзабіліті та конверсію. Він використовується компаніями з різних галузей, від малого бізнесу до великих корпорацій, оскільки має широкий спектр унікальних функцій.

Основні типи даних, які збирає Hotjar:

- Теплові карти – це візуалізація даних про те, де користувачі натискають, клацають та проводять курсором миші на сторінці. Приклад такої карти показаний на рисунку 1.3.
- Записи сеансів – це відеозаписи, як користувачі взаємодіють із вашим сайтом.

- Опитування та форми зворотного зв'язку - дані, зібрані з відповідей на опитування, анкети та форми зворотного зв'язку.
- Звіти про використання форм - аналіз взаємодії з формами, включаючи метрики, такі як час заповнення, місця відмов.
- Воронки конверсії — це інструмент для відстеження того, як користувачі проходять через ваш сайт, від входу до здійснення цільової дії.

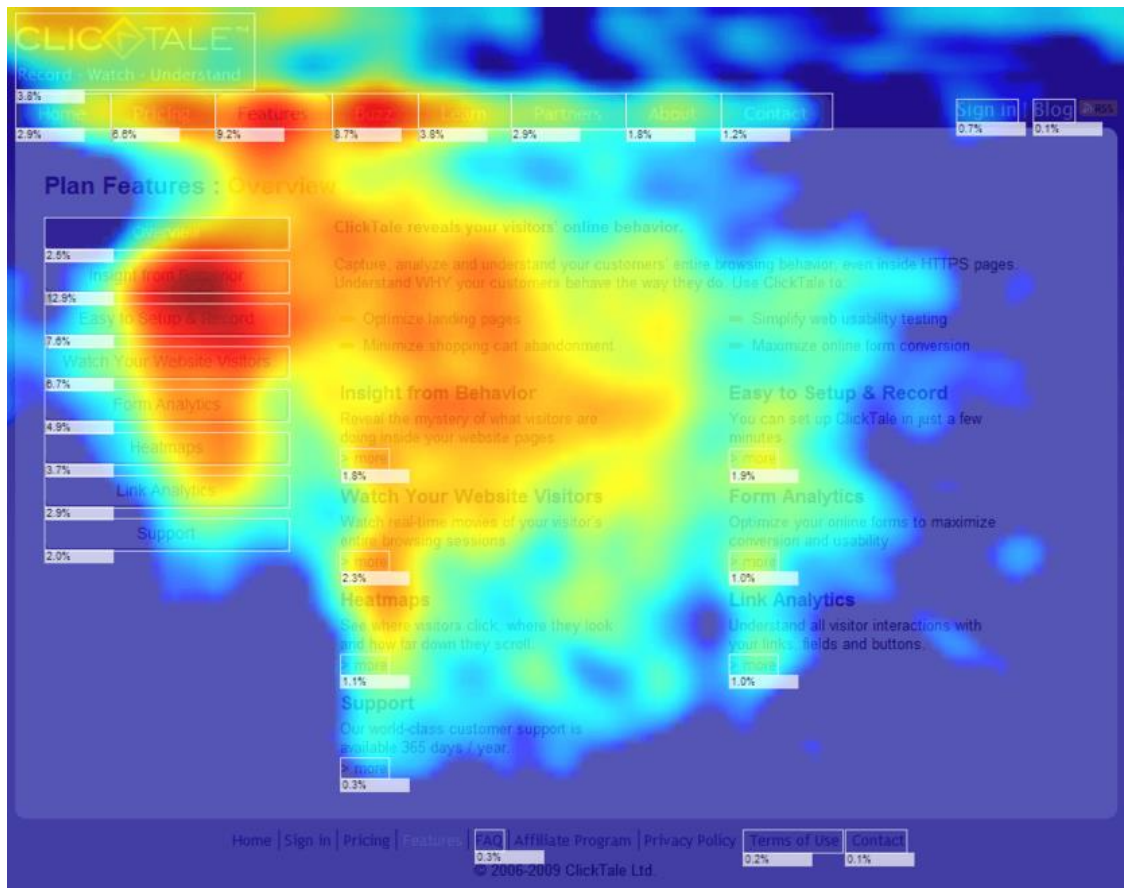


Рисунок 1.3 – Теплова карта

Hotjar надає досить точну інформацію, але слід зауважити деякі важливі моменти. Щоб Hotjar ефективно збирав дані, необхідно правильно вбудувати код відстеження на веб-сайт. Неправильні налаштування можуть призвести до втрати даних або викривлення результатів.

Деякі дані, такі як опитування або форми зворотного зв'язку, можуть залежати від активної участі користувачів. Якщо користувачі не відповідають або не беруть участь у опитуваннях, це може вплинути на точність даних.

Недоліки Hotjar:

1. Проблеми з конфіденційністю та безпекою. Записи сеансів можуть містити чутливу інформацію, таку як особисті дані користувачів. Необхідно вживати запобіжних заходів для забезпечення безпеки цих даних, щоб відповідати законодавству про конфіденційність.
2. Вплив на продуктивність. Використання запису сеансів може вплинути на продуктивність веб-сайту, особливо під час високого трафіку. Це може стати проблемою для користувачів з повільним з'єднанням або для пристроїв з обмеженими ресурсами.
3. Обмежений аналіз мобільних додатків. Інструмент переважно орієнтований на аналіз веб-сайтів, і його можливості аналізу мобільних додатків обмежені.

### **1.4.3 Adobe Analytics**

Adobe Analytics - це комплексне рішення, яке надає широкий набір функцій, що дозволяють збирати, аналізувати і візуалізувати дані про активність користувача отримані з веб-сайту, мобільного додатка та інших цифрових каналів.

Adobe Analytics збирає різноманітні дані, щоб надати комплексне уявлення про взаємодію користувачів із цифровими ресурсами. Ось кілька категорій даних, які зазвичай збираються за допомогою даного інструменту:

- Відвідування та користувачі: унікальні відвідувачі та нові користувачі, тривалість сесії та частота візитів, глибина перегляду сторінок та показники залучення.
- Джерела трафіку: інформація про те, звідки приходять користувачі (пошукові системи, реклама, реферальні посилання, прямий трафік, соціальні мережі).
- Параметри пристрою та технічні характеристики.



- Географічна інформація: розташування користувачів по країнах, регіонах та містах, мова браузера та переваги, пов'язані з локальними налаштуваннями.
- Події та взаємодія: відстеження подій, таких як кліки, прокручування сторінок, надсилання форм.
- E-commerce: дані про продаж, включаючи кількість проданих товарів, виручку, середній чек, інформація про товари, додані в кошик, та завершення покупок.
- Сегментація користувачів: створення сегментів користувача на основі різних параметрів (поведінкові, географічні, історії взаємодії і т.д.).
- A/B тестування – дані про продуктивність різних варіантів контенту чи дизайну. Наочно приклад A/B тестування показаний на рисунку 1.4.

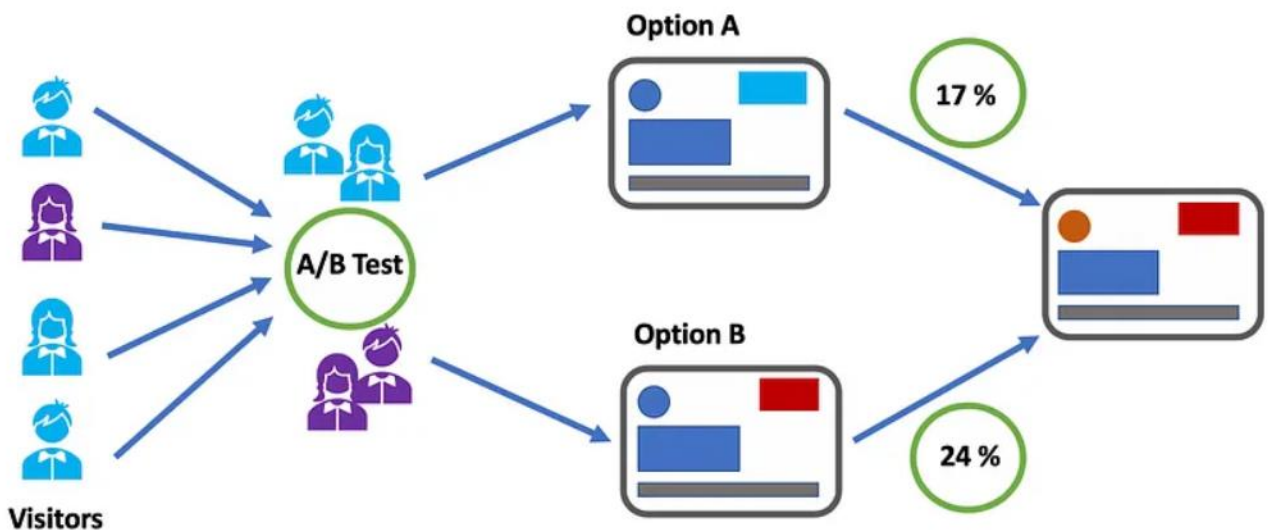


Рисунок 1.4 – A/B тестування

Adobe Analytics надає можливості для виявлення та фільтрації дублікатів та ботів, що є великою перевагою та важливо для підтримки чистоти даних та запобігання спотворенням результатів.

Цей інструмент підтримує аналіз даних у багатовимірному контексті, що дозволяє користувачам розглядати дані з різних точок зору та отримувати глибоке розуміння впливу різних параметрів на метрики.

### Недоліки Adobe Analytics:

1. Складність використання. Адміністраторам може знадобитися час для того, щоб освоїти всі можливості та налаштувати систему відповідно до їхніх потреб.
2. Складнощі в інтеграції. Інтеграція Adobe Analytics з іншими системами та інструментами може вимагати додаткових зусиль. Це може бути викликано різними форматами даних, складністю налаштування або необхідністю додаткових налаштувань.

Обмеження на рівні безпеки даних. У зв'язку з тим, що Adobe Analytics є хмарним сервісом, деякі компанії можуть мати обмеження щодо безпеки даних і віддають перевагу аналогічним інструментам, розгорнутими всередині їх інфраструктури.

Розглянувши зведену таблицю 1.1 з перевагами та недоліками розглянутих систем UBA, приходимо до висновку, що вибір сервісу слід робити, враховуючи встановлені вимоги до системи.

Таблиця 1.1 – Порівняння функціональних можливостей систем UBA

Функціональна можливість	Google Analytics	Hotjar	Adobe Analytics
Відстеження трафіку	+	+	+
Візуалізація досвіду користувача	Обмежено	+	Обмежено
Відстеження конверсій	+	+	+
Звіти та аналітика	+	Обмежено	+

Продовження таблиці 1.1

Функціональна можливість	Google Analytics	Hotjar	Adobe Analytics
Рівень складності використання	Середній	Низький	Високий
Сегментація користувачів	+	+	+
A/B тестування	-	-	+
Теплові карти	-	+	-
Стеження за подіями	Обмежено	-	+
Підтримка мобільних додатків	+	-	+

#### 1.4.4 Індивідуальна система UBA

Індивідуальна система UBA представляє собою власне рішення, спеціально розроблене для конкретної фізичної або юридичної особи. Часто ІТ-компанії створюють такі системи для особистого використання, орієнтуючись на інтеграцію власного рішення в розроблювані індивідуальні проекти.

Основна перевага приватної системи UBA - це її адаптивність до конкретних потреб клієнта. Компанія може обрати оптимальний набір функцій та технології, а також розробити власну бізнес-логіку. Це дозволяє адаптувати продукт до змінних умов бізнесу.

Хоча розробка та підтримка індивідуальної системи аналізу та моніторингу активності користувача вимагає додаткових витрат на початку, ці інвестиції окупуються в довгостроковій перспективі. Ваша система стає унікальним інструментом, що точно відповідає потребам вашого бізнесу.

Приватні системи UBA надають найвищий рівень безпеки. Це забезпечується тим, що власник має повний контроль над усіма аспектами системи, починаючи від збору, зберігання даних та закінчуючи їх обробкою. Цей високий рівень контролю не тільки зміцнює безпеку конфіденційної інформації, а й дозволяє активно реагувати на потенційні загрози, забезпечуючи захист користувачів на найвищому рівні.

Оскільки така система розробляється з урахуванням конкретних потреб бізнесу, вони точно конфігуруються під вимоги, що забезпечує ефективніше використання ресурсів. Така оптимізація дозволяє як підвищити продуктивність системи, так і мінімізувати витрати, що є додатковою перевагою для бізнесу.

Отже, якщо бізнесу потрібна система з високим ступенем адаптації до потреб бізнесу, високою швидкістю та масштабованістю, а також безпекою та конфіденційністю даних, то розробка власної системи є оптимальним рішенням.

## 2 ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР ТЕХНІЧНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

### 2.1 Постановка задачі щодо розробки універсального модуля UBA

Переді мною була поставлена така задача: створити систему аналізу та моніторингу активності користувачів, яка буде забезпечувати швидку пропускну здатність, кросплатформеною, матиме здатність легко масштабуватись і може бути використана незалежно від кінцевого призначення програмного продукту.

Після проведеного дослідження існуючих реалізацій UBA, було встановлено, що наявним продуктам не вистачає таких відмінних особливостей:

- Здатність до аналізу багатовимірних даних. Системи аналізу та моніторингу часто стикаються з величезними обсягами даних. Однак, багато з них не мають достатньої потужності для ефективного аналізу багатовимірних даних, включаючи різні джерела інформації, що обмежує можливості виявлення складних взаємозв'язків.
- Адаптивність до особливостей проекту. Система повинна бути гнучкою та легко налаштовуватися для підтримки різних сценаріїв використання та врахування специфічних особливостей програмного продукту.
- Масштабованість та продуктивність. Зі збільшенням числа користувачів та обсягу даних система має залишатися продуктивною. Масштабованість є критичним фактором, щоб забезпечити ефективне функціонування в умовах постійної активності користувачів.
- Інтелектуальний аналіз даних. Існуючі рішення зазвичай надають базовий аналіз користувальницької активності, але їхня здатність до глибокого та інтелектуального аналізу даних залишається обмеженою.

Система UBA створена з урахуванням виділених особливостей, створить відповідні умови для обробки великого потоку даних та універсального використання. Такий продукт обіцяє стати високо цінованим та затребуваним.

## 2.2 Вибір підходящих технологій для розробки універсального модуля UBA

Будучи ключовим етапом у життєвому циклі проекту, вибір технологій потребує уважного аналізу та компетентного прийняття рішень, щоб встановити стійкий фундамент. Технологічний стек визначає не тільки поточні можливості проекту, а і його розвиток в майбутньому. Рішення, прийняті на етапі вибору технологій, можуть суттєво вплинути на продуктивність, масштабованість та обслуговуваність системи у довгостроковій перспективі. При виборі технологій необхідно враховувати специфіку проекту, його цілі та вимоги. Також, важливо оцінити сумісність технологій. Отже, тільки правильний вибір технічного стеку допоможе досягти успішності та надійності проекту.

Для вирішення задачі було використано такий стек технологій:

- мова програмування Java;
- фреймворк Spring Boot;
- база даних MongoDB;
- брокер повідомлень Kafka;
- бібліотека Weka;
- збирач проектів Gradle.

### 2.2.1 Мова програмування Java

Java – це високорівнева, об'єктно-орієнтована, строго типізована, платформонезалежна мова програмування загального призначення, яка широко використовується в корпоративному середовищі.

Ключовою частиною екосистеми Java є JVM. Вона забезпечує абстракцію між програмою та апаратним забезпеченням, що дозволяє проектам на Java бути платформонезалежними. Коли програма компілюється, вона не транслюється в машинний код конкретної апаратної платформи. Натомість створюється байт-код,

який є проміжним представленням. JVM інтерпретує цей байт-код в машинний код, який може бути виконаний конкретним процесором. Наочно принцип роботи JVM зображено на рисунку 2.1.

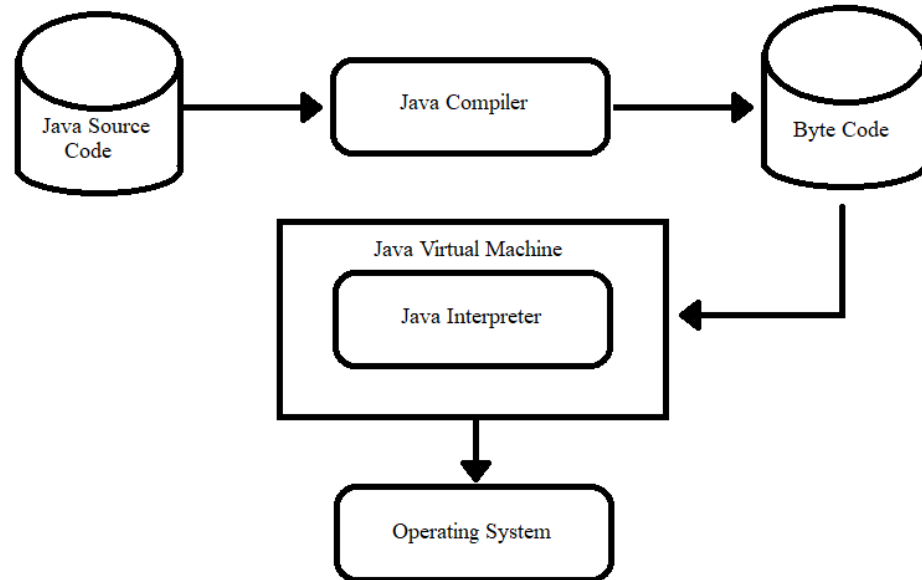


Рисунок 2.1 – Принцип роботи JVM

Завдяки портованості, яку забезпечує JVM, розробники можуть створювати системи, які легко масштабуються на різних пристроях та серверах. Якщо програма стає занадто великою або складною для роботи на одному пристрої, можна легко розгорнути на декількох пристроях або серверах. Це дозволяє програмам зростати і розвиватися зі зростанням потреб бізнесу.

Java є однією з найбезпечніших мов програмування у світі. Вона має низку вбудованих функцій безпеки, які допомагають захистити програми від атак зловмисників. Ось деякі з основних причин, чому варто використовувати Java з точки зору безпеки:

- Програми виконуються всередині JVM, яка ізолює їх від операційної системи та інших програм. Це ускладнює доступ до системи та даних програми для зловмисників.
- У Java відсутній прямиий доступ до вказувників та ручне керування пам'яттю. Це знижує ризик уразливостей, таких як переповнення буфера та витоку пам'яті.

- Ця мова програмування є строго типізованою, що означає використання системи перевірки типів даних, що використовуються в коді. Це допомагає запобігти помилкам, які можуть бути використані зловмисниками для отримання контролю над додатком.

Java має вбудовану підтримку багатозадачності і багатопоточності, що дозволяє ефективно використовувати ресурси системи. Багатозадачність Java реалізується за допомогою механізму потоків, що дозволяє виконувати кілька завдань одночасно в межах однієї програми. Кожен потік може працювати незалежно, що сприяє паралельній обробці завдань та збільшує загальну продуктивність програми. Це особливо важливо у розробці високопродуктивних та масштабованих додатків, де ефективне використання ресурсів стає критичним чинником.

### **2.2.2 Фреймворк Spring Boot**

Spring Boot – це фреймворк для розробки додатків на мові Java. Він надає зручні засоби для створення автономних програм, які можуть легко запускатися, конфігуруватися та розгортатися. Фреймворк надає готові рішення для поширених завдань, майже ідентичних для кожного проекту. Це звільняє розробників від рутини, дозволяючи їм зосередитися на написанні коду, який є унікальним для поточного проекту.

Spring Boot базується на платформі Spring Framework, що забезпечує широкий спектр можливостей розробки додатків, включаючи інверсію управління (IoC), ін'єкція залежностей (DI), аспектно-орієнтоване програмування (AOP) та інші.

Інверсія керування означає зміну того, як керування потоком виконання переходить від вашого коду до зовнішнього фреймворку або контейнера. Як показано на рисунку 2.2, замість того, щоб ваш код керував потоком виконання, як це зазвичай буває, контейнер IoC перехоплює цей контроль, інвертуючи його.



Ін'єкція залежностей - конкретний спосіб реалізації IoC, це процес забезпечення залежностей компоненту ззовні, а не створення їх усередині самого компонента. Таким чином, DI допомагає зробити компоненти більш незалежними, повторно використовуваними та полегшує тестування.

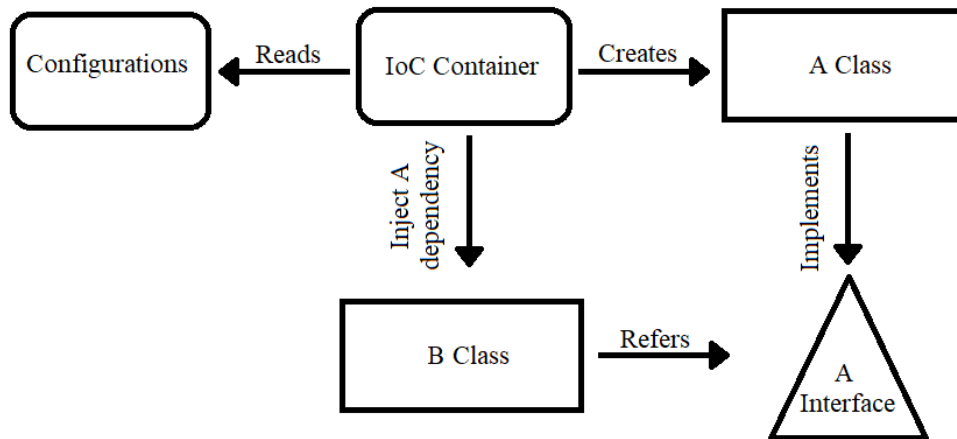


Рисунок 2.2 – Принцип роботи IoC

Spring Boot включає вбудовані сервери додатків, наприклад Tomcat. Це означає, що програми можуть бути запуснені незалежно від наявності встановленого сервера програм, що усуває необхідність вручну налаштовувати та запускати сервер. Крім того, Spring Boot використовує концепцію автоматичного конфігурування, що дозволяє додатку налаштовуватися без необхідності явно визначати конфігурацію.

Spring Boot легко інтегрується з багатьма бібліотеками та фреймворками, що спрощує взаємодію з базами даних, системами повідомлень, бібліотеками автентифікації та іншими технологіями.

### 2.2.3 База даних MongoDB

MongoDB – це документоорієнтована NoSQL-система управління базами даних, що означає, що вона не використовує традиційні реляційні таблиці, а натомість зберігає дані у форматі BSON (бінарний JSON) у документах, які об'єднані в колекції.

Ця база даних не вимагає попереднього визначення схеми для даних. Це означає, що кожен документ колекції може мати різні поля, їх структура може бути змінена в будь-який час. Це дозволяє працювати з неструктурованими даними та підлаштовуватися у процесі розробки, що особливо важливо, якщо формати та типи даних можуть змінюватися з часом. Гнучка схема дозволяє легко адаптуватися до змін вимог.

MongoDB підтримує створення індексів, які допомагають ефективно знаходити та отримувати необхідні записи, скорочуючи час виконання запитів та покращуючи загальну продуктивність операцій читання. Використання індексів дозволяє уникнути повного сканування колекції під час фільтрації даних, що скорочує використання ресурсів та прискорює виконання запитів. Крім того, інструменти для виконання складних операцій агрегації, таких як групування, фільтрація та перетворення даних, дають запитам ще більше функціональності та гнучкості.

MongoDB надає можливість горизонтального масштабування шляхом розподілу даних кількома серверами. Це допомагає забезпечити високу доступність та обробку великих обсягів даних та забезпечувати високу продуктивність навіть при збільшенні навантаження. Механізм реплікації забезпечує високу доступність і стійкість до відмови бази даних шляхом створення дублікатів даних на декількох серверах.

## **2.2.4 Брокер повідомлень Apache Kafka**

Apache Kafka – це розподілена платформа для стримінгової обробки даних та обміну повідомленнями між різними програмами та системами в реальному часі. Цей інструмент використовується в різних галузях, де потрібна надійна та масштабована платформа для обробки потоків даних, включаючи аналітику, моніторинг, обробку подій та багато іншого.

Продюсери та консьюмери є ключовими компонентами системи. На рисунку 2.3 показано принцип роботи Apache Kafka. Продюсери відповідальні за

ініціювання передачі повідомлень у топіки, представлені у брокері Kafka. Топіки є своєрідним механізмом категоризації даних, полегшуючи організацію та управління потоком інформації. У свою чергу, консьюмери підписуються на зазначені топіки, очікують на появу нових повідомлень і обробляють їх. Кожна група консьюмерів має можливість читати дані з топіків паралельно, що сприяє розподіленій обробці повідомлень і підвищує пропускну здатність системи.

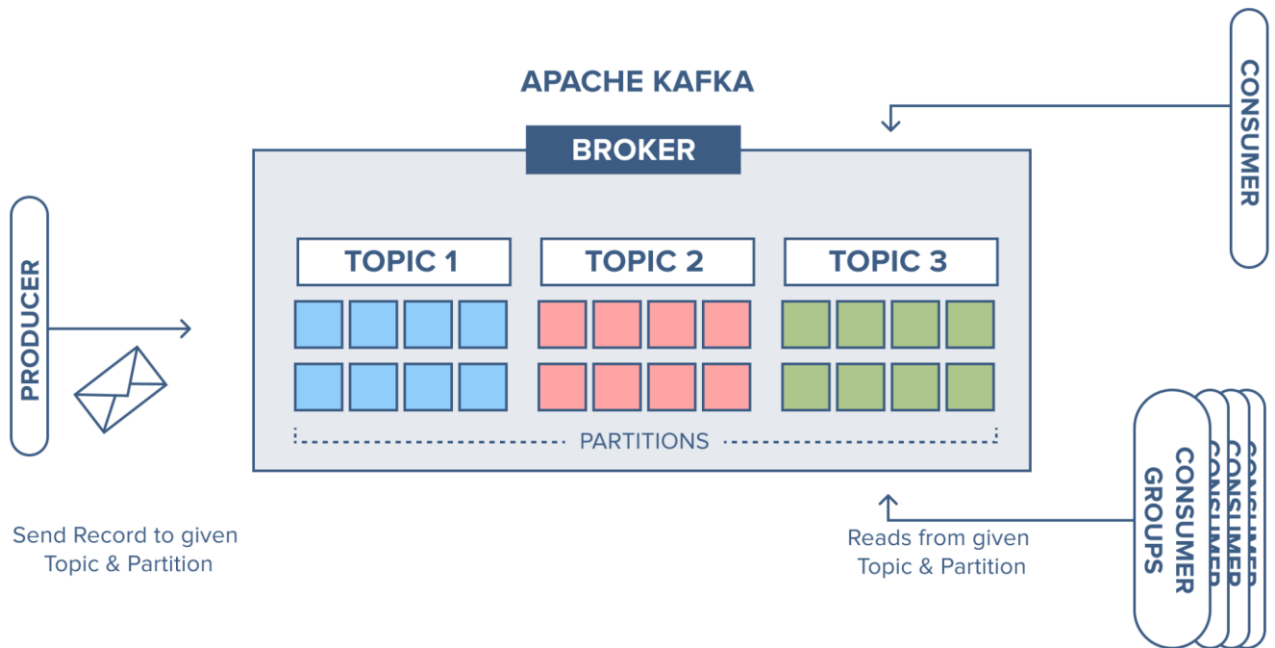


Рисунок 2.3 – Принцип роботи Apache Kafka

Повідомлення в Kafka зберігаються в топіках, які являються логічними групами повідомлень. Повідомлення у топіках зберігаються в порядку їх надходження, що дозволяє відновити їх у хронологічному порядку. Кожен топік у Kafka може мати кілька реплік, і ці репліки розподілені в різних брокерах в кластері. Коли повідомлення надходить у топік, воно спочатку записується на одному з брокерів, який є лідером цього топіка. Лідерський брокер потім реплікує це повідомлення іншим брокерам-реплікам. Таким чином забезпечується надійність та доступність повідомлень навіть при відмові окремих вузлів системи. Наочно це показано на рисунку 2.4.

Kafka забезпечує високу масштабованість як вертикальну, так і горизонтальну. Горизонтальне масштабування є привілейованим методом масштабування Kafka для більшості систем. Воно дозволяє збільшити пропускну

здатність та ємність кластера без необхідності збільшення апаратних ресурсів кожного брокера, просто необхідно додати нові брокери до кластера. Kafka автоматично управляє балансуванням між брокерами та партиціями, перерозподіляючи дані для забезпечення рівномірного навантаження.

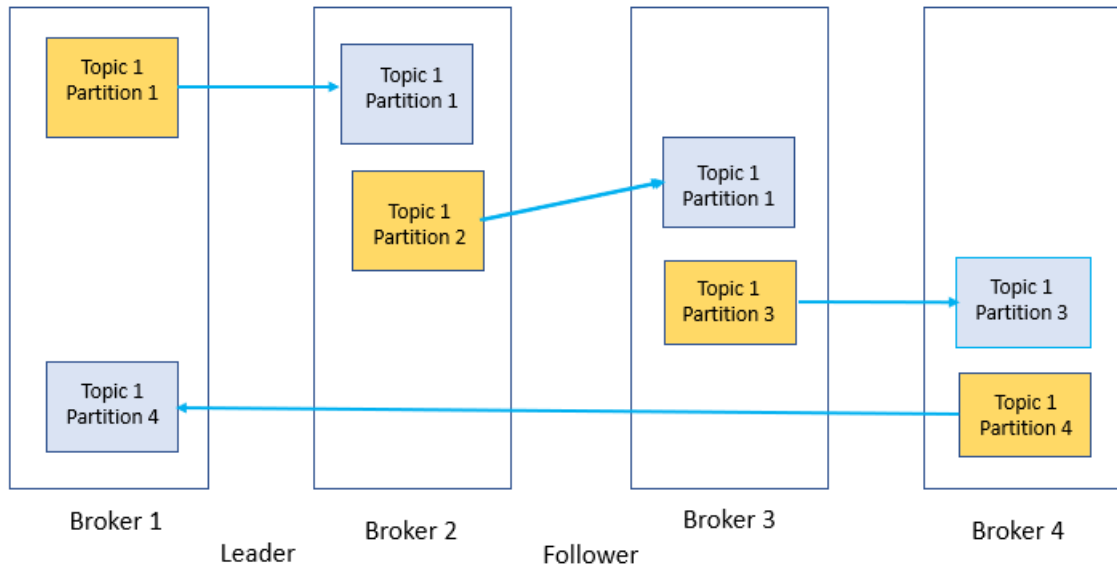


Рисунок 2.4 – Реплікація Apache Kafka

### 2.2.5 Бібліотека Weka

Weka – це написана на Java бібліотека, яка надає набір засобів для аналізу даних та машинного навчання. Вона надає широкий набір інструментів для вирішення різних завдань, включаючи:

- Безліч алгоритмів машинного навчання. Містить понад 100 алгоритмів машинного навчання, ці алгоритми можуть використовуватися для задач класифікації, регресії, асоціації, кластеризації та інших видів аналізу даних.
- Інструменти попередньої обробки даних. Weka надає безліч інструментів для попередньої обробки даних, таких як перетворення даних, очищення даних, масштабування, вибір ознак, обробка пропущених значень та інше. Це дозволяє готувати дані для навчання моделей.

- Інтерфейс візуалізації даних. Має графічний інтерфейс користувача, що робить даний інструмент зручним для використання та експериментів з різними методами машинного навчання. Це особливо корисно для дослідження даних, виявлення патернів та прийняття рішень щодо вибору моделі. Однак також може бути використана у вигляді бібліотеки в кодї Java.
- Підтримка кількох форматів даних, що дозволяє легко інтегрувати дані із різних джерел.
- Інструменти оцінки моделей для визначення ефективності моделей машинного навчання.

Weka є потужною бібліотекою машинного навчання, розробленою на мові Java, що забезпечує їй переносимість і, таким чином, працює на будь-якій сучасній обчислювальній платформі. Бібліотека проста у використанні та надає об'ємну документацію. Підтримка різних форматів даних та безліч інструментів візуалізації роблять Weka підходящим вибором для широкого кола завдань у галузі обробки даних.

### **2.2.6 Збірник проєктів Gradle**

Gradle - це інструмент для автоматизації збирання проєктів, призначений для управління залежностями, виконання тестів, генерації документації та багато чого іншого. Gradle добре інтегрований з екосистемою JVM і може ефективно взаємодіяти з іншими інструментами та бібліотеками цієї екосистеми.

Інструмент був розроблений для багатопроєктних збирань, які можуть досягати великих розмірів. Він працює на основі серії завдань збирання, які можуть виконуватися послідовно або паралельно. Gradle базується на концепції "конвенції замість конфігурації". Це означає, що Gradle використовує набір стандартних правил для побудови проєкту, а розробник може налаштувати ці правила відповідно до своїх потреб. Gradle пропонує підтримку всіх етапів процесу збирання, включаючи компіляцію, перевірку, вирішення залежностей, виконання

тестів, генерацію вихідного коду, упаковку та публікацію. Принцип роботи Gradle наочно показаний рисунку 2.5.

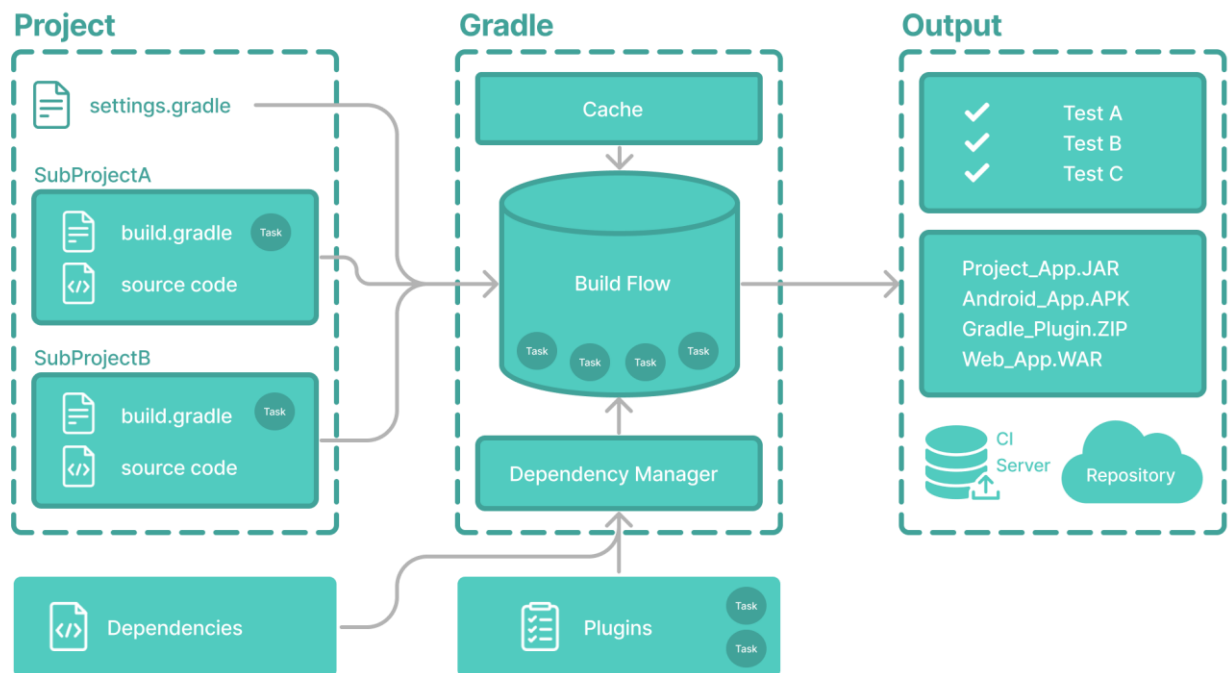


Рисунок 2.5 – Принцип роботи Gradle

Gradle представляє проект як графу завдань. Завдання – це одиниця роботи, яка виконується Gradle. Завдання можуть бути простими, наприклад компіляція коду, або складними, наприклад, запуск тестів або розгортання програмного забезпечення. Збирач використовує інтелектуальний механізм збирання, який дозволяє виконувати лише необхідні завдання, підтримує кешування компонентів збирання. Це допомагає значно прискорити процес збирання.

Gradle забезпечує зручний механізм управління залежностями. Ви можете визначити залежності у вашому збиральному скрипті, і Gradle автоматично перевірить їхню наявність і, за необхідності, завантажить їх із зазначених репозиторіїв.

Значна перевага даного збирача - це його гнучкість. Gradle дозволяє розробнику налаштувати процес збирання відповідно до своїх потреб. Це може бути корисним для складних проектів, які вимагають нестандартних складальних процесів.

Gradle підтримує багатопроєктні збирання, що дозволяє керувати кількома проєктами в одному репозиторії. Крім того, цей інструмент надає інкрементальне складання, що означає, що при зміні коду будуть перезібрані ті частини проєкту, які дійсно змінилися.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ СИСТЕМИ UBA

### 3.1 Розробка діаграми прецедентів

Якісна та повна специфікація вимог є основою для успішного виконання поставленої задачі. Діаграма прецедентів дає загальне уявлення про те, як різні типи користувачів взаємодіють з системою, але не включає в себе глибокого вдавання у деталі.

Система UBA має 2 актори:

1. Система-споживач.
2. Адміністратор.

Як показано на рисунку 3.1, системі-споживач відкриті такі прецеденти:

- відправлення даних активності користувача;
- отримання рекомендацій;
- запуск розсилки про важливі події;
- отримання результатів аналізу.

Відправлення даних активності користувача - це процес, який ініціює система-споживач, щоб передати дані про дії користувача. Це може відбуватися в результаті виконання певної події, наприклад, натискання на кнопку, завершення покупки та інше. Цей процес включає збирання інформації про результати опитувань, незвичайні події, які не реєструються системою UBA автоматично, та інші контекстуальні відомості. Отримані дані обробляються, формуються відповідно до вимог системи UBA.

Отримання рекомендацій - це процес отримання рекомендацій, сформованих на основі аналізу зібраних та наданих даних про активність з використанням різних алгоритмів та методів аналізу. Цей прецедент включає отримання загальних рекомендацій у ситуації, коли даних недостатньо, а також отримання рекомендацій для конкретного користувача, що надає основу для створення персоналізованих пропозицій.



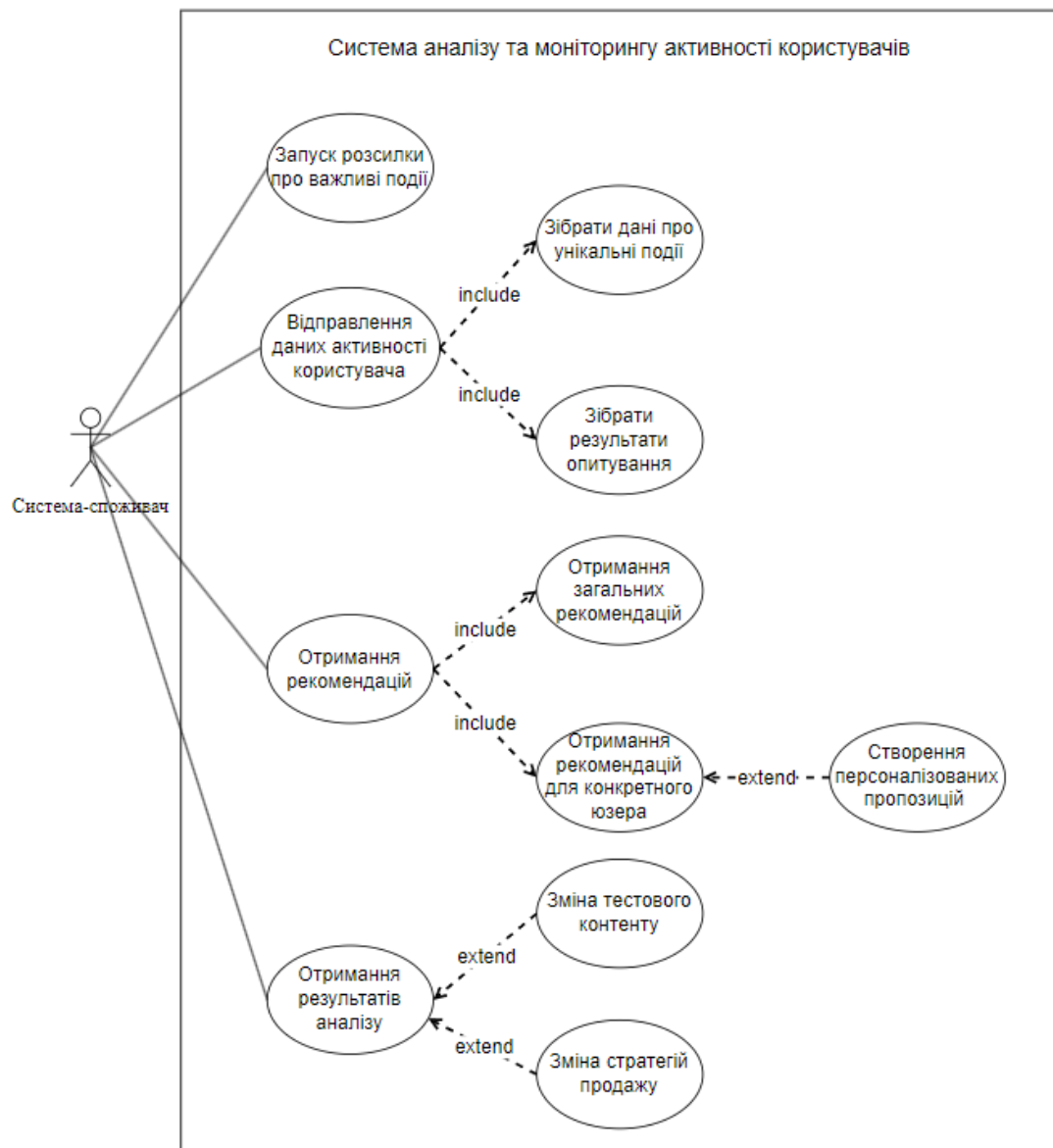


Рисунок 3.1 – Діаграма прецедентів системи UBA для системи-споживача

Запуск розсилки про важливі події - це процес розсилки повідомлень користувачам, що може бути частиною стратегії маркетингу чи комунікації з клієнтами, наприклад, залучення давно неактивних користувачів або надання спеціальних пропозицій активним. Після завершення розсилки, система UBA може почати моніторинг її результатів. Ця інформація може бути використана для подальшого аналізу та оптимізації стратегії розсилок.

Прецедент отримання результатів аналізу є ключовим у системі UBA. Його завдання - надати дані, що відображають реакцію користувачів на зміни в системі, та створити основу для прийняття стратегічно важливих рішень щодо заміни тестового контенту та стратегій продажу. Отримані висновки можуть виявити

неефективність поточного тестового контенту, що може спричинити його заміну. Вони також можуть виявити патерни та тенденції у поведінці користувачів, що можуть вимагати коригування поточної стратегії продажу.

Як показано на рисунку 3.2, адміністратору відкриті такі 3 прецеденти:

- моніторинг активності;
- конфігурація системи;
- створення звітності;
- резервне копіювання даних;
- отримання повідомлень про аномалії.

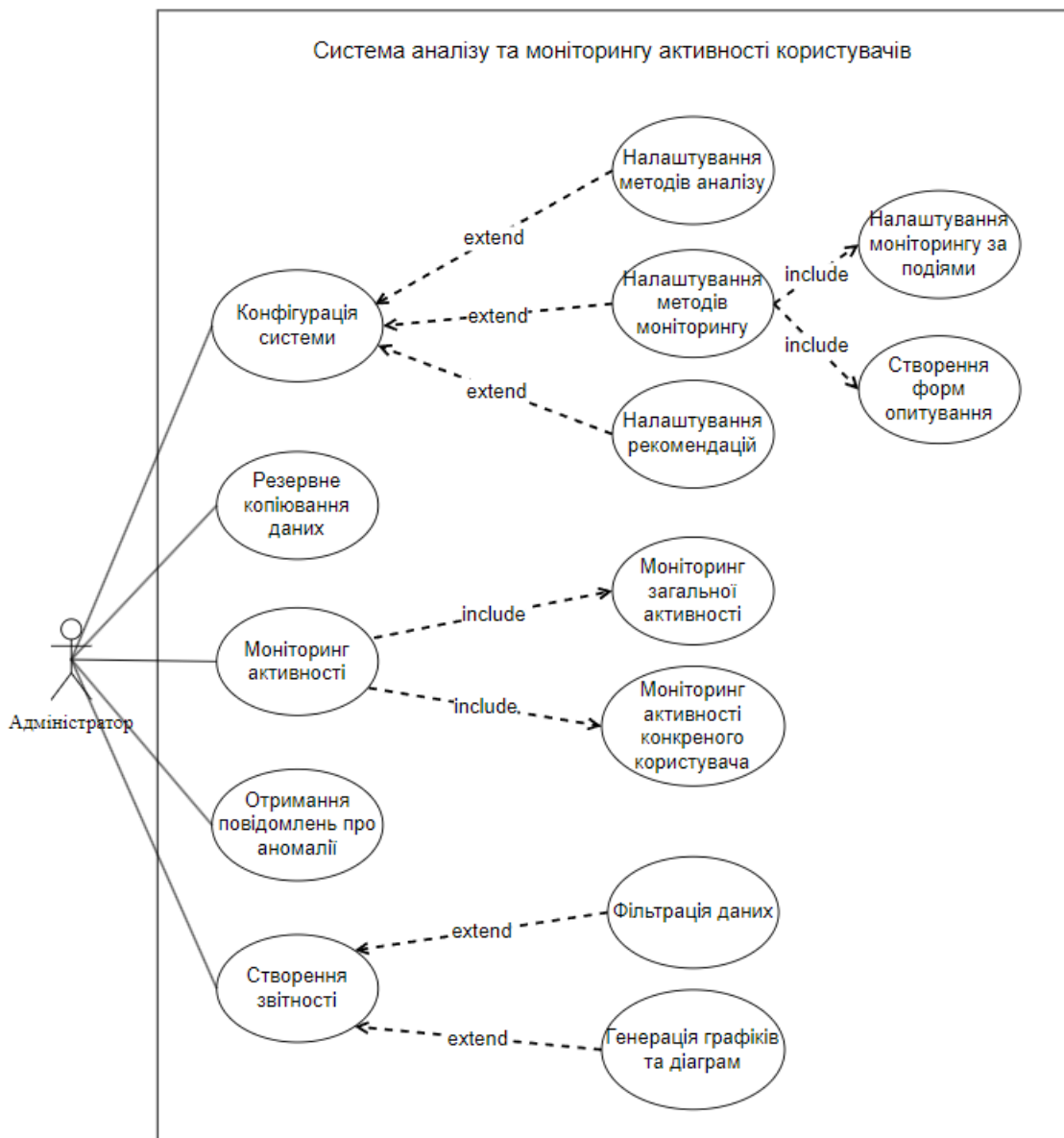


Рисунок 3.2 – Діаграма прецедентів системи UBA для адміністратора

Моніторинг активності - це процес, який дозволяє адміністратору стежити за діями користувачів у системі з ціллю отримання вичерпної інформації, наприклад даних про відвідані сторінки, час активності, скоєні дії, використання ресурсів. Цей прецедент включає моніторинг як загальної активності, так і конкретного користувача. Адміністратор має можливість вибирати різні параметри, такі як період часу, типи активності, користувачі та інше.

Конфігурація системи - це процес, у рамках якого адміністратор системи здійснює налаштування різних аспектів функціональності, виходячи з потреб системи-споживача.

Адміністратор може розширити цей прецедент, щоб додати або змінити методи аналізу даних. Це може включати визначення нових алгоритмів аналізу, налаштування параметрів існуючих методів або впровадження додаткових інструментів для обробки даних.

У рамках конфігурації системи є можливість розширити процес, щоб також внести зміни до методів моніторингу. Це може включати створення нових або редагування існуючих форм опитувань, вибір конкретних подій для відстеження, налаштування порогових значень для оповіщень або додавання нових джерел даних для більш повного моніторингу.

Налаштування рекомендацій дозволяє адміністратору визначити правила та параметри, на основі яких система генеруватиме рекомендації, які елементи інформації включатиме в рекомендації, і як користувач буде проінформований про них. Частина процесу конфігурації може включати установку метриків для оцінки ефективності рекомендацій. Це дозволяє адміністратору аналізувати, наскільки успішно система надає рекомендації та вносити відповідні корективи.

Створення звітності – це процес, який адміністратор системи виконує для отримання корисної інформації із зібраних даних. Цей прецедент включає кілька розширень:

- Фільтрування даних. Адміністратор визначає критерії фільтрації даних, щоб виділити необхідну інформацію із загального обсягу. Застосування

різних фільтрів може включати вибір конкретного часового періоду, певних користувальницьких груп або інших параметрів.

- Генерація графіків та діаграм. Надають візуальне уявлення складних даних, роблячи їх більш зрозумілими та доступними для аналізу. Візуалізація дозволяє швидко виявити тренди, аномалії або особливості активності користувача, які могли б бути втрачені при розгляді сирих числових даних.

Резервне копіювання даних - це збереження актуальної копії даних системи для забезпечення можливості відновлення у разі втрати або пошкодження, що відіграє ключову роль у забезпеченні надійності даних. Це допомагає уникнути втрати цінних даних у разі збоїв обладнання, програмних помилок, вірусних атак чи інших надзвичайних ситуацій. Система надає адміністратору можливість вибору типу даних для резервного копіювання.

Отримання повідомлення про аномалії – це процес інформування адміністратора про потенційні проблеми або аномалії, які можуть вимагати його своєчасного втручання. Система надсилає повідомлення заздалегідь налаштованими каналами зв'язку, наприклад електронна пошта, SMS. Важливою метою цього прецеденту є забезпечення оперативності реагування на аномалії.

### **3.2 Проектування структури системи UBA**

Проектування системи – це важливий етап розробки, а саме процес створення концептуальної моделі системи, яка визначає її структуру. Проектування системи необхідне для того, щоб забезпечити її відповідність вимогам користувачів та бізнес-цілей. Це відіграє ключову роль у створенні ефективних, стійких та легко супроводжуваних програмних продуктів. Крім того, розуміння архітектури та взаємозв'язків компонентів допомагає швидше реагувати на зміни вимог або вносити оновлення без суттєвих порушень функціональності.

Спроекуємо серверну частину системи аналізу та моніторингу активності користувача, так як саме серверна частина відповідає на запити, що надходять від

клієнтів, містить основну логіку програми і виконує різні операції, які визначають функціональність.

Щоб спростити складні системи, можна приховати подробиці реалізації та демонструвати лише основні функції. Такий підхід також сприяє модульності.

Модульна архітектура - це підхід до проектування та побудови програмних систем, у якому функціональність системи розбивається на невеликі, незалежні компоненти, які називаються модулями. Ці модулі виконують конкретні завдання та взаємодіють один з одним за чітко визначеними інтерфейсами. Розподілення системи на дрібніші, керовані модулі допомагає знизити її складність, покращити зручність обслуговування та можливості повторного використання. Це робить систему гнучкішою і забезпечує легкість її підтримки та подальшого розвитку.

Модулі повинні бути максимально незалежними один від одного і організовані так, щоб кожен з них виконував конкретну функцію або вирішував певне завдання. Це дозволяє легше розуміти, тестувати та змінювати кожен модуль незалежно від інших.

Модульна архітектура полегшує повторне використання коду, оскільки кожен модуль може бути розроблений і протестований незалежно, а потім використаний у різних частинах системи або навіть в інших проектах.

Добре спроектовані модулі повинні мати чіткі межі та мінімізувати залежності між ними. Таким чином, якщо потрібно модифікувати функціональність, можна зосередитись на відповідних модулях, мінімізуючи вплив на інші частини системи. Модулі можна легко додавати, видаляти або замінювати, не торкаючись інших модулів системи. Це дозволяє швидко адаптувати систему до змінних вимог.

Розглянемо структуру серверної частини системи аналізу та моніторингу активності користувача на рисунку 3.3.

Система UBA надає зовнішнє API, яке є інтерфейсом, що дозволяє системі-клієнту взаємодіяти з системою аналізу та моніторингу без знання внутрішньої реалізації, забезпечуючи легкість інтеграції. Крім того, система може обробляти

запити від різних клієнтів, включаючи веб-програми, мобільні програми та інше. Це дуже корисно, оскільки багато сервісів мають відразу кілька клієнтів.

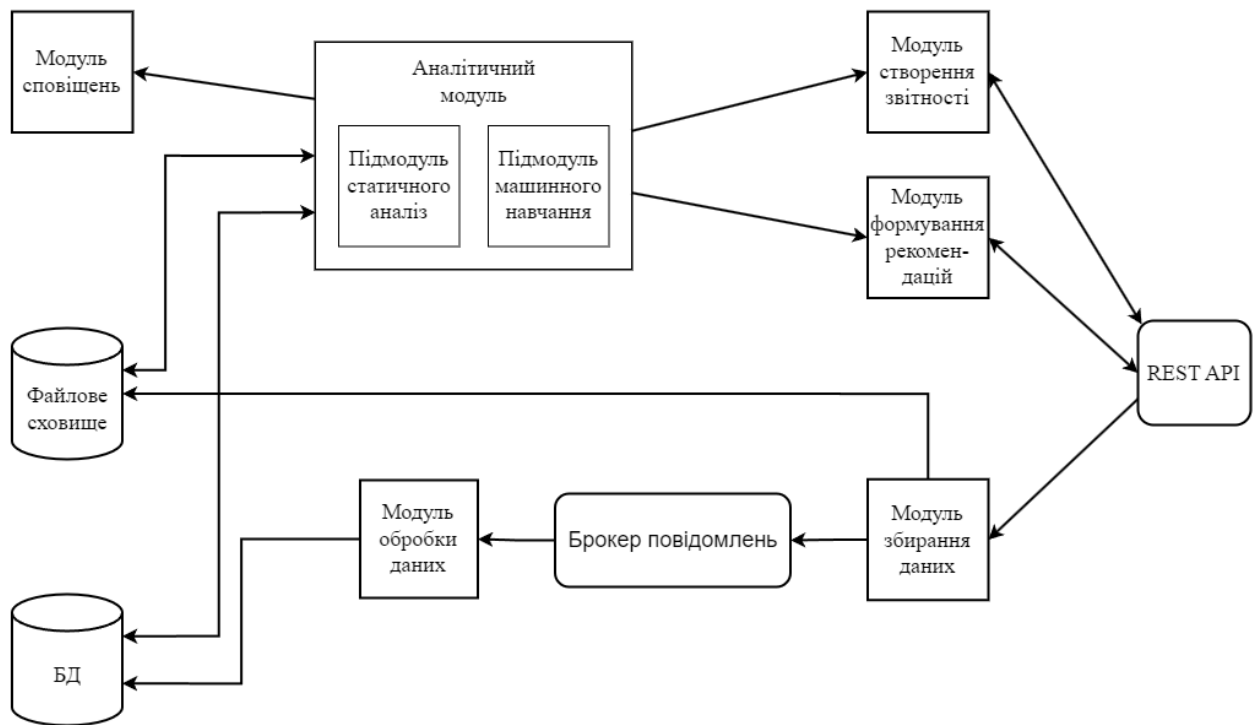


Рисунок 3.3 – Структура системи UBA

Надання API дозволяє гнучкіше керувати правами доступу та забезпечувати безпеку даних. Система регулює, які дані та функціональності доступні через API, контролюючи рівні доступу.

Модуль збору даних призначений для систематичного збору, агрегації та збереження інформації про дії користувачів. Надані через API дані фільтруються і надалі або надсилаються на брокер повідомлень, або зберігаються у файлому сховищі, залежно від типу даних. Наприклад, лог-файли будуть збережені, а дані про кліки будуть надіслані на брокер.

Брокер повідомлень є ключовим компонентом в архітектурі. Його роль полягає у забезпеченні ефективного та надійного обміну інформацією між різними модулями. Брокер повідомлень дозволяє розподілити навантаження та відокремити процеси, що виконуються різними модулями системи. Кожен модуль може функціонувати незалежно, надсилаючи та приймаючи повідомлення через брокера.

Основна перевага, яку надає брокер повідомлень - це можливість виконувати обробку повідомлень асинхронно. Таким чином, модуль збору даних передає повідомлення і не чекає відповіді, а далі продовжує виконувати роботу. Модулі можуть обмінюватись повідомленнями без необхідності знати про існування один одного.

Брокер повідомлень забезпечує надійність доставки повідомлень, а також підтримувати різні стратегії обробки помилок. Навіть у разі збою одного з модулів інші модулі зможуть далі отримувати та надсилати дані.

Модуль обробки даних отримує дані з брокера повідомлень, потім виконує завдання фільтрації та попередньої обробки, щоб отримувати необхідну інформацію, а також приводити дані до однакового формату, що дозволяє ефективніше працювати з різними типами інформації. Нормалізація включає приведення дат і часу, встановлення загальних форматів даних, заповнення пропущених значень, кодування категоріальних змінних та інше. Також модуль відповідає за перевірку цілісності даних, виявлення помилок та їх корекцію. Це допомагає уникнути проблем, пов'язаних з помилковими або пошкодженими даними. Після обробки дані зберігаються у базі даних.

Аналітичний модуль розділений на підмодулі статичного аналізу і машинного навчання. Цей модуль отримує дані з бази даних. Якщо під час аналізу виявляється якась аномалія, то аналітичний модуль надсилає сповіщення системному адміністраторові, використовуючи модуль сповіщень, який обирає спосіб доставки та надсилає повідомлення. Це забезпечує оперативне реагування на потенційні загрози або проблеми в системі.

Підмодуль статичного аналізу використовує математичні методи для виявлення закономірностей і отримання кількісних оцінок різних аспектів. Підмодуль машинного навчання має різноманітні навчені моделі, які використовуються для різних необхідних завдань аналітики, передбачення, виявлення уподобань та інше.

Модуль формування рекомендацій використовує отримані результати з аналітичного модуля, враховує задані налаштування уподобань, створює та

відправляє персоналізовані рекомендації контенту, в залежності від специфіки системи-клієнта.

Модуль створення звітності є гнучким та налаштовуваним, щоб користувачі могли формувати звіти, що відповідають їх потребам. Даний модуль використовує отримані результати з аналітичного модуля, враховує задані параметри, фільтри та створює звіти. Також створює графіки, діаграми та інші засоби візуалізації для полегшення сприйняття інформації. Підтримує різні формати, щоб дати можливість експортувати звіти у зручному для адміністратора форматі.

### **3.3 Розробка модуля збору даних**

Універсальний модуль аналізу та моніторингу активності користувача повинен надавати кілька способів передавати та збирати дані, щоб і десктопне програмне забезпечення, і мобільні додатки, і веб-додатки могли взаємодіяти з ним. Використання HTTP-запитів та брокера повідомлень є найбільш універсальними способами, які можуть бути використані для збору даних про активність з різних систем.

#### **3.3.1 Збір даних через HTTP-запити**

HTTP – це широко розповсюджений протокол передачі довільних даних, який використовується для обміну інформацією між клієнтом та сервером у мережі Інтернет. HTTP-запит є повідомленням, що надсилається клієнтським додатком (наприклад, веб-браузером, мобільним додатком) серверу з метою виконання певної дії.

HTTP-запит складається з наступних частин:

1. URI – ідентифікатор ресурсу, до якого звертається клієнт.
2. Заголовок запиту - додаткова інформація про запит, наприклад тип вмісту, мова, куки, ключ для авторизації та інші параметри.



3. Тіло запиту - необов'язкова частина запиту, що містить дані, які потрібно надіслати серверу. Дані можуть бути в будь-якому форматі, найпоширенішими є JSON та XML. Заголовок Content-Type вказує тип вмісту в тілі, наприклад, application/json.

HTTP визначає безліч методів запиту, які вказують, яку дію потрібно виконати для певного ресурсу. Найбільш поширеними методами запиту є:

- GET – отримати інформацію про ресурс.
- POST – створити новий ресурс.
- PUT – замінити існуючий ресурс.
- PATCH – частково змінити ресурс.
- DELETE – видалити ресурс.

У відповідь на HTTP-запит сервер відправляє HTTP-відповідь, яка складається з наступних частин:

- Рядок статусу - статус обробки запиту, що вказує на успішність виконання запиту або наявність помилок, наприклад "200 OK", "400 Bad Request", "401 Unauthorized", "500 Internal Error" та інші.
- Заголовок відповіді - метадані, що надають додаткову інформацію про передані дані та сервер.
- Тіло відповіді - необов'язкова частина відповіді, що містить фактичні дані, які були запитані або можуть містити інформацію про невдале виконання запиту.

Але у HTTP є один істотний недолік, який можна помітити, якщо перехопити відправлений запит - дані передаються у відкритому вигляді, HTTP сам по собі не надає ніяких засобів шифрування. HTTPS є розширенням HTTP-протоколу, яке використовує шифрування для забезпечення конфіденційності даних. Дані передаються в єдиному зашифрованому потоці, що унеможливорює отримання облікових даних користувачів та іншої критичної інформації засобами звичайного перехоплення.

HTTPS-запити відрізняються від HTTP-запитів лише тим, що вони передаються захищеним каналом зв'язку. В іншому їх структура та обробка аналогічні HTTP-запитам. HTTPS є обов'язковим для всіх веб-сайтів, які збирають або обробляють конфіденційні дані користувачів.

Модуль аналізу та моніторингу користувальницької активності надає RESTful API, що робить його більш доступним для розробників, забезпечує простоту інтеграції та взаємодії з різними програмами та платформами. Відмінною особливістю сервісів REST є те, що вони дозволяють якнайкраще використовувати протокол HTTP.

REST — архітектурний стиль взаємодії компонентів розподіленої програми у мережі. Він заснований на принципі представлення ресурсів та передачі стану між клієнтом та сервером. RESTful системи прагнуть бути простими, легкими у використанні та підтримувати прозорість взаємодії між компонентами в мережі. REST описує узгоджений набір принципів, які роблять веб-сервіси масштабованими, гнучкими, стійкими та забезпечують окремість між клієнтом та сервером.

Існує п'ять обов'язкових обмежень для побудови розподілених REST-додатків:

1. Модель клієнт-сервер. Клієнт та сервер взаємодіють один з одним через мережу, клієнт надсилає запит серверу, а сервер повертає відповідь. Це дозволяє окремим частинам розвиватися незалежно друг від друга.
2. Відсутність стану. Сервер не зберігає клієнта. Кожен запит сприймається як окремий. Це дозволяє серверу обробляти запити від кількох клієнтів одночасно й масштабуватись.
3. Кешування. Зберігає копії даних у локальній пам'яті для прискорення доступу до них. Ресурси повинні містити інформацію про те, чи можна кешувати подання ресурсів чи ні. Правильне використання кешування може скоротити час, необхідний для обробки запитів від клієнтів, що підвищить продуктивність системи та її здатність обробляти більше запитів.

4. Одноманітність інтерфейсу. У REST-архітектурі всі ресурси мають спільний інтерфейс. Це означає, що клієнти можуть взаємодіяти з ресурсами, незалежно від того, де вони знаходяться або як вони реалізовані. Уніфіковані інтерфейси дозволяють кожному із сервісів розвиватися незалежно.
5. Шари. Архітектура може бути поділена на шари, де кожен шар виконує певні функції. Кожен шар не знає про те, що відбувається за його межами та відповідає за певну функцію. Це дозволяє спростити розробку та обслуговування системи, а також здатне підвищити масштабованість за рахунок розподіленого кешування та балансування навантаження.

Якщо сервіс-додаток порушує будь-яку з цих умов, цю систему не можна вважати REST-системою.

RESTful API – це веб-сервіс, що реалізує принципи архітектури REST. Такий інтерфейс прикладного програмування надає стандартизований спосіб взаємодії між клієнтом та сервером за допомогою передачі уявлень ресурсів. RESTful API зазвичай використовує стандартні методи HTTP для виконання операцій над ресурсами, а також підтримує використання форматів даних, таких як JSON або XML. Розробники можуть створювати як клієнтські, так і серверні програми різними мовами програмування, не торкаючись структури API. Також можна змінити базову технологію на будь-якій стороні, не впливаючи на обмін даними.

Для нашого RESTful API клієнтами є десктопні програми, мобільні додатки, веб-додатки, системи повідомлень та інші зовнішні сервіси, а ресурсами є користувачі, події, дані активності, метрики, звіти та статистика, конфігурації, повідомлення та інше.

Опишемо кілька запитів збору даних:

- POST /api/user/{id}/session-duration – приймає дані про тривалість сеансу користувача.
- POST /api/user/{id}/page-views- приймає дані про переглянуті користувачем сторінки.

- POST /api/user/{id}/actions - приймає дані про конкретні дії користувача на сайті або у додатку.
- POST /api/user/{id}/purchases - приймає дані про здійснені користувачем покупки.
- POST /api/user/{id}/profile-changes - приймає дані про внесені користувачем зміни до свого профілю.
- POST /api/user/{id}/search-queries - приймає дані про пошукові запити.
- POST /api/user/{id}/location - приймає дані про географічне розташування користувачів, якщо це дозволено користувачами.
- POST /api/user/{id}/preferences - приймає дані про уподобання користувача.
- POST /api/user/{id}/survey - приймає дані про результати опитувань, пройдених користувачем. Включає обрані відповіді та іншу пов'язану інформацію про проходження опитування.

Процес нормалізації даних перед відправкою до системи аналізу та моніторингу є важливим етапом для забезпечення правильного та ефективного аналізу даних. Перед надсиланням даних клієнтський додаток повинен виконати валідацію даних, уніфікувати структуру, формат та значення даних, що полегшує їх інтерпретацію та порівняння.

Вибираємо JSON як формат даних для взаємодії між клієнтами та сервером, оскільки JSON є стандартизованим і часто використовується у веб-розробці, що спрощує інтеграцію з веб-додатками; зазвичай має менший обсяг даних, порівняно з іншими форматами; надає простий та легкий для читання формат даних, що спрощує налагодження, тестування та взаємодію як для розробників, так і для систем; легко серіалізується та десеріалізується, що означає, що його можна легко перетворити на рядок або об'єкт. Також ми використовуємо MongoDB, який зберігає JSON-подібні документи, тому нам не потрібно буде зайвий раз переводити формати.

Проаналізуємо приклад тіла запиту від системи-клієнта на рисунку 3.4. Так як ми використовуємо JSON, легко зрозуміти призначення змінних. Зокрема, `user_id` представляє ідентифікатор користувача, `activity_type` містить інформацію про тип активності, `timestamp` вказує тимчасову мітку активності, `details` містить додаткові відомості, такі як IP-адреса, браузер та операційна система. Отже, система-клієнт передає дані про авторизацію конкретного користувача.

```
{
  .... "user_id": "5f5b88f3b2b52c2d68a99634",
  .... "activity_type": "login",
  .... "timestamp": "2023-01-01T12:34:56",
  .... "details": {
    ..... "ip_address": "192.168.1.1",
    ..... "browser": "Chrome",
    ..... "platform": "Windows"
  }
}
```

Рисунок 3.4 – Тіло запиту, що містить дані про авторизацію користувача

### 3.3.2 Збір даних через брокер повідомлень

Брокер повідомлень - це програмний сервіс або посередник, який забезпечує надсилання повідомлень між різними компонентами програми або системи. Він відіграє важливу роль в архітектурі розподілених систем, де компоненти програми можуть знаходитись на різних серверах або пристроях. Цей метод особливо корисний у контексті того, що наш модуль буде використаний іншими системами-клієнтами, де події активності користувача можуть бути відправлені в реальному часі. Брокер повідомлень забезпечує надійну доставку даних, а також дозволяє легко масштабувати систему за потреби.

Основне завдання брокера повідомлень - забезпечувати асинхронний обмін даними між компонентами, дозволяючи їм взаємодіяти незалежно від розташування та часу роботи. Це дозволяє надсилати повідомлення без очікування відповіді, розділити процеси та обробляти їх у фоновому режимі, що зменшує

залежності між компонентами та забезпечує більш гнучку та масштабовану архітектуру.

Продюсерами є системи-клієнти, які використовують систему аналізу та моніторингу. Продюсери відправляють інформацію про активність в брокер повідомлень, який зберігає повідомлення, що надійшли, забезпечуючи надійність передачі даних навіть у разі тимчасових недоступностей компонентів системи. Також, при необхідності, брокер може виконувати попередню обробку даних, наприклад фільтрацію.

Брокер повідомлень надає механізм, який полегшує публікацію та підписку на повідомлення, ефективно організовує обмін повідомленнями між різними компонентами та називається топіком. Кожне повідомлення, відправлене до топіка, буде доставлене всім консьюмерам, які зацікавлені в цьому топіку.

Консьюмерами є аналітичні модулі, які підписуються на необхідні топіки. При надходженні нових даних брокер повідомляє відповідні модулі, дозволяючи їм оновлювати свої результати аналізу та надавати інформацію в реальному часі. На рисунку 3.5 наведений консьюмер, який підписаний на топік, що містить інформацію про кліки. Консьюмер отримує та обробляє дані, а в кінці надсилає підтвердження про успішну обробку. Це дозволяє брокеру повідомлень позначити повідомлення як успішно отримане. У випадку помилки підтвердження не буде відправлене, тож інший консьюмер його опрацює натомість.

```
@KafkaListener(groupId = "userClickGroup", topics = "userClicks")
public void consumeUserClicks(@Payload String message, Acknowledgment acknowledgment) {
    //Обробка повідомлення, значення вилучено з тіла Kafka-повідомлення
    userActivityHandler.handle(message);
    //Надсилаємо підтвердження про успішну обробку повідомлення
    acknowledgment.acknowledge();
}
```

Рисунок 3.5 – Консьюмер, що підписаний на топік з інформацією про кліки

Опишемо кілька топіків, в які записуються зібрані дані:

- UserClicks - записи про кліки користувача з різних елементів інтерфейсу.
- UserSearches - інформація про пошукові запити, введені користувачами.

- UserSessions - дані про сесії користувачів, такі як початок, кінець та тривалість сесії.
- UserPurchases - інформація про здійснені користувачами покупки та транзакції.
- UserPageViews - дані про перегляд сторінок, включаючи інформацію про час перегляду.
- UserSocialActivity - дані про соціальну активність користувачів, такі як шеринг контенту та взаємодія в соціальних мережах.
- UserLocation - інформація про розташування користувачів.
- UserPreferences - інформація про уподобання користувачів.
- UserSurveys - результати опитувань та анкет, які проводяться для збору думки користувачів.

Топіки можуть поділятися на партиції, які використовуються для покращення продуктивності, розподілу навантаження та забезпечення масштабованості. Кожна партиція може розглядатися як окрема послідовність чи потік подій. Якщо система не встигатиме обробляти велику кількість даних в якомусь топіку ми можемо збільшити кількість партицій, відповідно і споживачів, і провести ребалансування для перерозподілу навантаження. Дане конфігурування не вдасться провести заздалегідь, потрібно орієнтуватися на завантаженість системи.

### **3.4 Розробка аналітичного модуля**

Система аналізу активності користувача зазвичай використовує методи аналізу даних залежно від конкретних цілей необхідного аналізу. Універсальна система повинна використовувати комбінацію методів для отримання більш повної картини закономірностей і тенденцій у поведінці. Вибираємо статистичні методи та методи машинного навчання, щоб отримати кількісну оцінку різних аспектів та виявити закономірності у даних.

### 3.4.1 Статистичні методи аналізу

Статичний метод аналізу даних - це метод, який використовується для опису даних, отримання загальних відомостей, виявлення закономірностей та тенденцій, а також для перевірки гіпотез. Такі методи ґрунтуються на використанні математичних методів. Статичний аналіз даних означає вивчення даних у нерухомому стані, тобто без урахування часу чи послідовності подій.

Розглянемо найбільш поширені статистичні методи аналізу даних.

Описова статистика – це розділ статистики, що займається узагальненням даних. Використовується для опису даних, щоб отримати уявлення про їх основні характеристики, такі як середнє, медіана, стандартне відхилення та інше.

Середнє - це середнє значення у наборі даних. Медіана - це значення, яке знаходиться посередині набору даних, відсортованого в порядку зростання або зменшення. Стандартне відхилення - це показник, який показує, наскільки широко розподілені значення набору даних. Ці показники дозволяють зрозуміти як розподілені дані та виявити закономірності у них.

Цей спосіб допомагає візуалізувати основні статистичні характеристики. У системі аналізу активності користувача описова статистика може використовуватися для створення зведених звітів, що містять основні метрики, такі як середній час перебування на сайті, кількість переглянутих сторінок, розподіл активності за часом.

Інференційна статистика – це розділ статистики, який займається висновками про сукупність на основі даних, отриманих із вибірки. Вона дозволяє робити узагальнення населення у цілому, передбачення про характеристики популяції виходячи з даних, зібраних з невеликої групи людей. Інференційна статистика може використовуватися в системі аналізу активності користувача для перевірки гіпотез про поведінку користувачів. Наприклад, можна використовувати для перевірки гіпотези про те, що користувачі різних вікових груп по-різному використовують сайт.



Аналіз часових рядів - це розділ статистики, який займається вивченням даних, які є послідовністю вимірювань, отриманих протягом певного періоду часу. Він використовується для виявлення закономірностей, таких як тренди, сезонність та циклічність, а також для прогнозування майбутніх значень.

Тренд - це довгострокова зміна даних. Він може бути позитивним (дані зростають) або негативним (дані знижуються). Сезонність - це зміна, що повторюється, в даних, що відбувається з певною періодичністю. Циклічність - це триваліша зміна даних, що може тривати кілька років і навіть десятиліть.

Даний аналіз можна використовувати для виявлення сезонних коливань у використанні сайту, виявити сезонні патерни або визначити, які події викликають пік активності.

Аналіз кореляції - це метод, який використовується для вимірювання взаємозв'язку між двома чи більше змінними. Кореляція може бути позитивною, негативною або нульовою. Позитивна кореляція свідчить про те, що змінні зростають чи зменшуються разом. Негативна кореляція вказує на те, що змінні зростають або зменшуються у протилежних напрямках. Нульова кореляція свідчить про те, що між змінними немає взаємозв'язку. Сила кореляції вимірюється коефіцієнтом кореляції. Чим ближче коефіцієнт кореляції до 1, тим сильніший зв'язок між змінними.

Дозволяє виявляти взаємозв'язки між різними видами активності користувача, наприклад можна використовувати аналіз кореляції для виявлення взаємозв'язку між часом, яке користувачі проводять на сайті, і кількістю зроблених покупок.

Аналіз регресії - це метод, який використовується для прогнозування значень однієї змінної на основі значень іншої змінної. Він заснований на припущенні, що між змінними існує взаємозв'язок та дозволяє побудувати модель, яка описує взаємозв'язок між змінними. Лінійна регресія - це простий тип регресії, який передбачає, що зв'язок між змінними є лінійним. Множинна регресія - це складніший тип регресії, що дозволяє враховувати вплив кількох змінних на залежну змінну.

Даний аналіз може використовуватися для прогнозування одного виду активності користувача на основі іншого, наприклад можна побудувати модель, що передбачає ймовірність конверсії після певної взаємодії з сайтом.

Аналіз дисперсії - це метод, що використовується для оцінки впливу одного або кількох факторів на змінну відгуку. Він вимірює наскільки дисперсія значень змінної відгуку у кожній групі відрізняється від загальної дисперсії. Цей метод використовується для порівняння середніх значень у різних групах. Він дозволяє визначити, чи є різниця між середніми статистично значимими значеннями. У системі аналізу користувальницької активності даний метод допомагає визначити наскільки відрізняється активність користувачів у різних групах, наприклад можна визначити різницю середнє значення часу, яке проводять на сайті користувачі з різних типів пристроїв.

Кластерний аналіз - це метод класифікації, що використовується для групування об'єктів на основі їх схожості. Він знаходить групи об'єктів, які схожі один на одного за певним набором показників. У системі аналізу дозволяє групувати користувачів зі схожими характеристиками активності у кластери. Це може бути корисним для персоналізації контенту або маркетингових стратегій для різних груп користувачів.

Деякий аналіз не потребує важких математичних обчислень, тому потрібний результат можливо отримати одразу з бази даних. Розглянемо рисунок 3.6, на якому показаний запит до MongoDB та результат його виконання.

Ціль запита – отримати найбільш та найменш відвідувані сторінки. Для цього ми виконуємо запит до колекції, що містить дані про перегляд сторінок (pageViews), сортуємо по кількості візитів (visits) та проведеного часу (timeSpent) на сторінці, виділяємо перший та останній запис, записуємо до змінних mostPopularPage і leastPopularPage. Таким чином нам не потрібно робити лишніх математичних обчислень на Java.

```

userActivity> db.pageViews.aggregate([
...   {
...     $sort: {
...       visits: -1,
...       timeSpent: -1
...     }
...   },
...   {
...     $group: {
...       _id: null,
...       mostPopularPage: { $first: "$pageName" },
...       leastPopularPage: { $last: "$pageName" }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       mostPopularPage: 1,
...       leastPopularPage: 1
...     }
...   }
... ])
[ { mostPopularPage: 'home', leastPopularPage: 'aboutUs' } ]

```

Рисунок 3.6 – Отримання найбільш та найменш відвідуваних сторінок з MongoDB

### 3.4.2 Методи машинного навчання

Машинне навчання - це підрозділ штучного інтелекту, який займається розробкою алгоритмів та моделей, що дозволяють комп'ютерам отримувати закономірності з даних та навчатися на основі досвіду. Замість явного програмування, де розробники вказують комп'ютеру конкретні кроки виконання завдання, у машинному навчанні використовується підхід, у якому комп'ютер навчається на даних і робить передбачення чи приймає рішення без явного програмування.

Реалізація технологій машинного навчання відбувається на базі статистичних, математичних та ймовірнісних методів – тобто комп'ютер оцінює кожен унікальний випадок, порівнюючи її з величезною базою даних, в якій містяться більш-менш схожі випадки. Алгоритми ML здатні виконувати складні завдання з великою швидкістю та точністю. Завдяки адаптивному характеру

машинного навчання воно відмінно підходить для сценаріїв, в яких дані постійно змінюються, властивості запитів або завдань нестабільні або написати код для вирішення фактично неможливо.

Використання ML в системі аналізу активності користувача дозволяє створювати більш інтелектуальні та адаптивні системи, здатні краще відповідати потребам користувачів і надавати більш якісний досвід взаємодії. Розглянемо ряд переваг, які забезпечують більш ефективне та точне розуміння поведінки:

- Ефективно обробляє величезні обсяги даних.
- Допомогає автоматично виявляти групи користувачів зі схожою поведінкою. Це дозволяє ефективніше проводити сегментацію аудиторії.
- Може автоматично виявляти аномальну чи незвичайну поведінку користувачів, що є корисним для виявлення потенційних випадків шахрайства.
- Здатне створювати моделі, які передбачають майбутні дії користувачів на основі їхньої попередньої поведінки.
- Може поліпшити точність аналітики, виявляючи приховані патерни в даних, які можуть бути втрачені за традиційних методів аналізу.
- Дозволяє створювати персоналізовані рекомендації та контент, враховуючи переваги та потреби кожного конкретного користувача.

Процес машинного навчання може бути трудомістким і зазвичай включає такі основні етапи:

1. Збір даних. Цей етап включає збір різноманітних даних, які будуть використовуватися для навчання моделі. Навчальні дані надають моделі інформацію про вхідні ознаки та відповідні правильні відповіді. Чим різноманітнішими, якіснішими та репрезентативними є дані, тим краще модель зможе навчитися, розпізнавати образці та здійснювати правильні прогнози на нових даних.
2. Підготовка даних. Дані часто вимагають попередньої обробки та очищення. Це включає видалення або виправлення помилок, нормалізацію

значень, перетворення даних у формат та інші маніпуляції для того, щоб дані були готові для навчання моделі. Дані зазвичай поділяються на навчальну, валідаційну та тестову вибірки. Це необхідно для оцінки узагальнюючої здатності моделі на нових даних.

3. Вибір моделі. На цьому етапі вибирається архітектура моделі, яка буде використовуватися для навчання і є алгоритмом або математичною функцією, яка перетворює вхідні дані у вихідні. Модель вибирається залежно від завдання та типу даних. Вона може бути лінійною, деревом рішень, нейронною мережею і так далі. Одна з ключових цілей машинного навчання – створення моделей, котрі здатні видавати точні передбачення нових даних, які раніше не застосовувалися в процесі навчання.
4. Навчання. Модель навчається на тренувальних даних, пристосовуючись до закономірностей в даних і будуючи алгоритм для вирішення задачі. Процес навчання полягає у підганянні моделі до навчальних даних. Модель аналізує дані, виявляє закономірності та коригує свої внутрішні параметри так, щоб мінімізувати помилку між прогнозами моделі та правильними відповідями. Замість явного програмування моделі одержують знання з даних та коригують свої параметри для досягнення продуктивності.
5. Оцінка та тестування. Після навчання моделі необхідно оцінити її точність та продуктивність на нових даних. Для цього використовується тестовий набір даних, який не використовувався для навчання моделі. Оцінка проводиться за допомогою метрик, що вимірюють точність, повноту та інші характеристики моделі. Це дозволяє оцінити, як модель справляється із завданням та визначити необхідність подальшого доопрацювання.
6. Розгортання моделі. Після того, як модель була визнана прийнятною, її розгортають для використання у реальних додатках.

Чим більше даних обробляє така модель і чим довше вона використовується, тим точнішими стають результати, завдяки постійній адаптації до нових обставин і змін у даних. Цей процес, відомий як безперервне навчання, дає змогу моделі

зберігати актуальність та ефективність у динамічному середовищі, забезпечуючи надійне функціонування протягом тривалого часу.

Більшість завдань можна розділити на два види машинного навчання: навчання з учителем та навчання без учителя. У цих методах “учителем” може виступати програміст, який задає правила і контролює роботу алгоритму, але це не є обов'язковим. У контексті машинного навчання “учителем” може бути будь-яке втручання людини у процес обробки інформації. В обох випадках алгоритму надаються вихідні дані, які він має проаналізувати та знайти в них закономірності. Головна відмінність між навчанням з учителем та навчанням без вчителя полягає у наявності чи відсутності наданих гіпотез, які потрібно перевірити чи підтвердити. Також є третій тип – навчання з підкріпленням, де модель навчається приймати рішення та виконувати дії у певному середовищі, з метою максимізувати певну нагороду чи накопичену корисність. Давайте розглянемо докладніше ці типи.

Навчання з учителем – це найпоширеніший тип ML, у якому модель навчається з урахуванням маркованих даних, де кожен приклад даних має відповідну цільову змінну чи мітку. Модель прагне знайти залежності та загальні закономірності між вхідними даними та відповідними вихідними мітками, щоб передбачати результати для нових, раніше не бачених даних. Приклади алгоритмів навчання з учителем включають лінійну регресію, метод опорних векторів, випадкові ліси та нейронні мережі.

Сильні сторони машинного навчання з учителем – простота та легкість структури. Така система корисна при прогнозуванні можливого обмеженого набору результатів, розподілу даних на категорії або об'єднанні результатів двох інших алгоритмів машинного навчання. Однак маркування мільйонів наборів даних є складним завданням. Прикладами завдань є класифікація, регресія.

Класифікація - визначення належності об'єкта до конкретного класу. Модель використовує ці дані для навчання класифікувати нові записи відповідно до їх мітки класів. Наприклад, модель можна використовувати для класифікації повідомлень як спаму або неспаму.

Регресія - прогнозування безперервної цільової змінної. За допомогою алгоритмів регресії створюється модель на основі значень, яка потім використовується для прогнозування. Регресійні дослідження допомагають одержати прогноз на майбутнє: ймовірний попит на продукт, показники продажу або попередню оцінку результатів кампанії.

Навчання без вчителя – це тип ML, у якому модель навчається на немаркованих даних без явної цільової змінної. Натомість, модель шукає приховані структури, патерни та групи в даних. Алгоритми навчання без вчителя використовуються для кластеризації даних, зниження розмірності, асоціативного аналізу та генерації нових ознак. Оскільки навчальні дані не вимагають маркування, налаштування просте. Ці алгоритми також можна використовувати для автоматичного очищення та обробки даних для подальшого моделювання. Обмеження цього методу в тому, що він не може дати точних прогнозів. З іншого боку, він може самостійно виділяти конкретні типи вихідних даних. Такий тип навчання добре підходить для завдань, в яких необхідно знайти закономірності в даних. Прикладами завдань є кластеризація та виявлення аномалій.

Кластеризація – групування подібних об'єктів усередині даних. Наприклад, кластеризація використовується для групування клієнтів за їх купівельними звичками, групування текстів за їх змістом або групування зображень за їх схожістю.

Виявлення аномалій - виявлення записів, які відрізняються від інших у наборі даних. Це може бути корисним для виявлення шахрайства або інших проблем.

Навчання з підкріпленням – це тип ML, в якому модель навчається взаємодіючи із середовищем та отримуючи зворотний зв'язок у вигляді нагороди чи штрафу. Модель самостійно приймає рішення та налаштовує свою поведінку на основі отриманої нагороди. Мета агента – навчитися вибирати оптимальні дії для максимізації кумулятивної винагороди. Навчання з підкріпленням добре підходить для завдань, у яких модель має приймати рішення у середовищі. Прикладом завдання може бути персоналізація контенту. Модель використовується для навчання пропонувати користувачам контент або послуги, які вони швидше за все

оцінять. Наприклад, модель можна використовувати для рекомендацій фільмів, музики або для визначення релевантних оголошень для користувачів.

Розглянемо як оптимізувати стратегію електронного маркетингу за допомогою машинного навчання. Суть – підвищити ефективність email-кампаній. Що потрібно спрогнозувати – цільову аудиторію для розсилок електронною поштою. Результати ми зможемо використовувати для зменшення рівня занепокоєння клієнтів, підвищення залученості та рентабельності маркетингових кампаній електронною поштою.

Математичні алгоритми машинного навчання працюють із великими обсягами даних і знаходять навіть приховані закономірності поведінки клієнтів. Ці закономірності не видно менеджерам і, більше того, іноді навіть клієнти про них не підозрюють.

Дані як про користувача, так і про повідомлення ми можемо використовувати для навчання з вчителем. Алгоритми розпізнавання шаблонів з навчанням припускають наявність історичної інформації, що дозволяє побудувати моделі статистичного зв'язку  $x \rightarrow y$ , де  $y$  - дії користувача, що спостерігаються,  $x$  - множина змінних, за допомогою яких передбачається пояснити мінливість змінної  $y$ .

Перед побудовою моделі необхідно зібрати та проаналізувати будь-яку доступну інформацію. Цінність результатів залежить від актуальності даних, у яких проводився аналіз. Перш ніж ми дізнаємося реакцію користувача на черговий лист, ми знаємо лише попередню історію його поведінки. Сформуємо множину змінних  $x$  - виділимо такі показники:

- вік користувача;
- тип повідомлення;
- час існування користувача у системі;
- відсоток відкриття листів у будні/вихідні;
- середня кількість листів одержуваних/відкритих користувачем на тиждень;
- активність користувача за 5/10/15 днів.



Важливо, щоб у вашому наборі даних було достатньо прикладів як для позитивних, так і негативних прогнозів.

Змінна відгук  $y$  приймає два значення:

- 0 – користувач не прочитав лист;
- 1 – користувач прочитав лист.

Модель повинна передбачати значення змінної відгуку за заданого значення параметрів  $x_i \in X$ ,  $i = \{1, \dots, n\}$ . Іншими словами, модель відповідатиме на таке запитання: чи прочитає користувач черговий лист чи ні? Виходить, що завдання зводиться до бінарної класифікації – алгоритм навчається передбачати належність об'єкта до однієї з двох категорій чи класів.

Ключовим моментом з яким дослідник стикається при розробці моделі, що навчається, є вибір оптимального алгоритму вилучення закономірностей. Для нашого завдання як математичну функцію візьмемо функцію логістичної регресії – один із найефективніших класифікаторів машинного навчання для завдань бінарної класифікації.

Логістична регресія – це метод, який використовується для моделювання ймовірності приналежності спостереження до певної категорії. Основна ідея полягає в тому, щоб зв'язати лінійну комбінацію вхідних ознак із ймовірністю приналежності до певного класу, використовуючи логістичну функцію. Проаналізувавши рисунок 3.7 робимо висновок, що логістична функція перетворює множину значень в значення від 0 до 1.

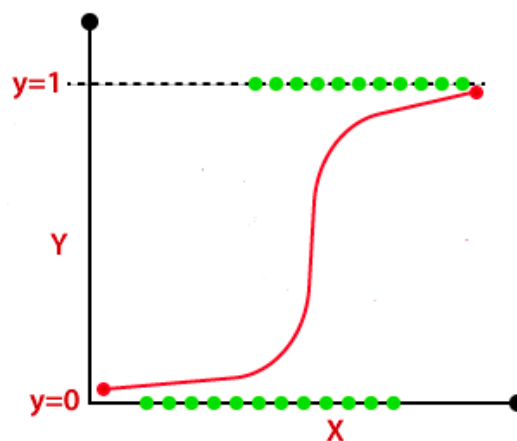


Рисунок 3.7 – Графік логістичної функції

Таким чином, при оцінці параметра у модель обчислює ймовірність прочитання листа, це дозволить не турбувати користувачів без необхідності. Використовуючи таку навчену модель компанія зможе збільшити результативність листів і при цьому скоротити загальну кількість листів, що відправляються. Якщо користувач рідше отримує листи, йому цікавіше читати їх. Також інший користувач більш імовірно вчинить цільову дію при отриманні чергового листа.

## ВИСНОВКИ

У першому розділі було визначено призначення та основні функціональні можливості системи аналізу та моніторингу активності користувачів, досліджено методи моніторингу та аналізу, проведена порівняльна характеристика існуючих систем.

У другому розділі були встановлені відмінні особливості яких не вистачає наявним системам, а саме здатність до аналізу багатовимірних даних, адаптивність до особливостей проекту, масштабованість та продуктивність, інтелектуальний аналіз даних. Вибраний та обґрунтований технічний стек, який забезпечує швидку пропускну здатність, кросплатформеність, здатність легко масштабуватись.

У третьому розділі були розроблені діаграми прецедентів для системи-клієнта та адміністратора, спроектовано структуру серверної частини системи та визначено архітектуру. Розроблено модуль збору даних, який використовує найбільш універсальні способи передачі даних, а саме HTTP-запити та брокер повідомлень. Розроблено аналітичний модуль, який використовує статистичні методи та методи машинного навчання для отримання кількісних оцінок, виявлення закономірностей.

В ході роботи були досліджені сучасні методи моніторингу та аналізу, які дозволяють в повній мірі збирати дані та проводити інтелектуальний аналіз. Розроблена система гарантує не тільки статистику використання програмного продукту, а й надає цінні данні для покращення досвіду користувача.

Отримані результати підтверджують перспективи та важливість застосування даного модуля в різних галузях, а також відкривають можливості для подальших досліджень та удосконалень у галузі аналізу та моніторингу активності користувача.

## ПЕРЕЛІК ПОСИЛАНЬ

1. User Analysis A Complete Guide - 2020 Edition / Gerardus Blokdyk - 5STARCooks, 2020- 300 с.
2. User activity monitoring A Complete Guide / Gerardus Blokdyk - 5STARCooks, 2021 - 314 с.
3. Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics (Interactive Technologies) - Morgan Kaufmann, 2013 - 320 с.
4. Web Analytics Comparison: Most Popular Web Analytics Tools in 2023 [Електронний ресурс] / Эксперте- Режим доступу: <https://www.experte.com/create-a-website/website-analytics>
5. The comparison of 10 web and app analytics platforms [Електронний ресурс] / Piwik- Режим доступу: <https://piwik.pro/blog/web-analytics-vendors-comparison-sheet/>
6. Spring in Action / Craig Walls - Manning, 2022 - 970 с.
7. MongoDB: The Definitive Guide, 3rd Edition / Shannon Bradshaw, Eoin Brazil, Kristina Chodorow - O'Reilly, 2019 - 426 с.
8. SQL & NoSQL Databases / Andreas Meier , Michael Kaufmann - Springer Vieweg, 2019 - 245 с.
9. Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale / Gwen Shapira, Todd Palino, Rajini Sivaram, Krit Petty - O'Reilly, 2022 - 488 с.
- 10.Spring for Apache Kafka [Електронний ресурс] / Spring. - Режим доступу: <https://spring.io/projects/spring-kafka>
- 11.WEKA v4.2 documentation [Електронний ресурс] / WEKA. - Режим доступу: <https://docs.weka.io/>
- 12.Introducing Gradle / Balaji Varanasi - Apress, 2015 - 148 с.
- 13.Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps / Eoin Woods - Addison-Wesley Professional, 2021- 325 с.
- 14.Practical Monitoring: Effective Strategies for the Real World 1st Edition / Mike Julian - O'Reilly, 2017 - 170 с.

15. Fundamentals of Software Architecture: A Comprehensive Guide to Patterns, Characteristics, and Best Practices 1st Edition / Neal Ford, Mark Richards - O'Reilly, 2020 - 550 с.
16. UML A Complete Guide - 2020 Edition / Gerardus Blokdyk — 5STARCOOKS, 2019 - 200 с.
17. HTTP [Електронний ресурс] / Вікіпедія. - Режим доступу:  
<https://uk.wikipedia.org/wiki/HTTP>
18. Machine Learning in Action / Peter Harrington - Manning, 2012 - 384 с.
19. Forecasting User Activity Using Machine Learning and R Language [Електронний ресурс] / Yespo- Режим доступу:  
<https://yespo.io/blog/forecasting-user-activity-using-machine-learning-and-r-language>
20. What is machine learning and how does it work? In-depth guide [Електронний ресурс] / TechTarget. - Режим доступу:  
<https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML>

# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

## КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Дослідження та розробка програмного модуля  
аналізу та моніторингу активності користувачів»

на здобуття освітнього ступеня магістра  
зі спеціальності 122 Комп'ютерні науки  
освітньо-професійної програми Комп'ютерні науки



Виконав:  
здобувач вищої освіти Роман ГОРКУН  
група КНДМ-61

Керівник: Сергій СЕРІХ  
доц, к.т.н

## Загальна характеристика кваліфікаційної роботи

### Наукове завдання

розробка серверної частини універсального модуля аналізу та моніторингу поведінки користувачів.

### Мета роботи

окреслити шляхи підвищення ефективності аналізу та збору даних про активність користувачів.

### Об'єкт дослідження

процес аналізу та моніторингу активності користувачів.

### Предмет дослідження

система аналізу та моніторингу активності користувачів.

## **Загальна характеристика кваліфікаційної роботи**

### **Наукове завдання**

розробка серверної частини універсального модуля аналізу та моніторингу поведінки користувачів.

### **Мета роботи**

окреслити шляхи підвищення ефективності аналізу та збору даних про активність користувачів.

### **Об'єкт дослідження**

процес аналізу та моніторингу активності користувачів.

### **Предмет дослідження**

система аналізу та моніторингу активності користувачів.

## **Загальна характеристика кваліфікаційної роботи**

### **Актуальність**

Поширення цифрових технологій породжують нові виклики та можливості для дослідження поведінки користувачів, щоб в подальшому використовувати цю інформацію у різних сферах, включаючи інтернет-маркетинг, покращення клієнтського досвіду та розвиток веб-ресурсів та додатків.

З точки зору бізнесу, продукт, правильно адаптований до потреб користувачів, забезпечує збільшення конверсії, збільшення продажів та залучення нових клієнтів. Підприємства, які вміють аналізувати та реагувати на активність користувачів, отримують конкурентну перевагу та можуть швидше підлаштовуватись під зміни в ринкових умовах.

Таким чином, використання програмного модуля для аналізу та моніторингу активності користувачів стає стратегічно важливою для будь-якої організації

## **Призначення та основні функціональні можливості систем UBA**

**Аналіз та моніторинг діяльності користувача** - це процес збору, обробки та аналізу даних про взаємодію користувачів з веб-системами, додатками, онлайн-платформами або іншими інформаційними системами з метою отримання інформації про їхню поведінку, зрозуміння їхніх потреб, вподобань та взаємодій з певними елементами програмного продукту.

### **Призначення**

- Поліпшення користувацького досвіду
- Підвищення ефективності роботи систем
- Прийняття рішень у галузі маркетингу та продажів
- Захист від шахрайства

### **Функціональні можливості**

- Збір та агрегація даних
- Аналіз поведінки користувачів
- Сегментація користувачів
- Реагування на зміни
- Персоналізовані рекомендації
- Візуалізація даних
- Системи оповіщень

## **Типи аналітики користувача – дескриптивна та сегментна аналітика**

### **Дескриптивна аналітика**

метод аналізу даних, який використовується для опису даних. Він не передбачає жодних висновків чи передбачень про дані, а просто надає інформацію про їх характер та структуру.

Аналіз починається із огляду загальної активності. Це включає виявлення основних метрик, таких як загальна кількість користувачів, кількість сеансів, динаміку зростання і падіння активності. Ці дані надають високорівневе уявлення про генеральну картину взаємодії клієнтів.

Це допомагає зрозуміти, які конкретні елементи чи послуги залучають користувачів найбільше.

### **Сегментна аналітика**

процес поділу користувачів на групи на основі їх загальної поведінки для забезпечення кращого досвіду.

Сегментна аналітика передбачає аналіз користувачів кожного сегмента споживачів.

Зазвичай сегментують користувачів на основі наступних моделей:

- демографічні показники;
- поведінка в програмі;
- обліковий запис;
- атрибути користувача;
- настрої користувачів.

Компанія може використовувати дані щодо поведінки користувачів, отримані в результаті аналізу сегментів, для надання індивідуальних рішень для кожного сегменту.



## Типи аналітики користувача – аналітика воронки

### Аналітика воронки

процес визначення користувачів, які конвертуються на кожному етапі воронки (ряд кроків, що використовуються для складання схеми руху користувачів до конверсії). Це дозволяє приймати обґрунтовані рішення для поліпшення досвіду користувача і, відповідно, коефіцієнта конверсії.

Таким чином, ви можете дізнатися, що стимулює конверсію, що змушує користувачів відторгатися до конверсії і оптимізує ефективність воронки.



## Типи аналітики користувача – когортна аналітика

### Когортна аналітика

дозволяє побачити, що когорта (підсекція користувачів) робить у вашому продукті. Компанії-розробники програмного забезпечення зазвичай використовують когортну аналітику для виміру відтоку клієнтів. Саме тому когортний аналіз називають аналізом відтоку клієнтів.

Залежно від вибраного типу, когортна аналітика допоможе відповісти на такі запитання:

- Який сегмент користувачів найчастіше відторгається?
- Як довго ви можете утримувати середньостатистичного клієнта і коли він залишає ваш продукт?

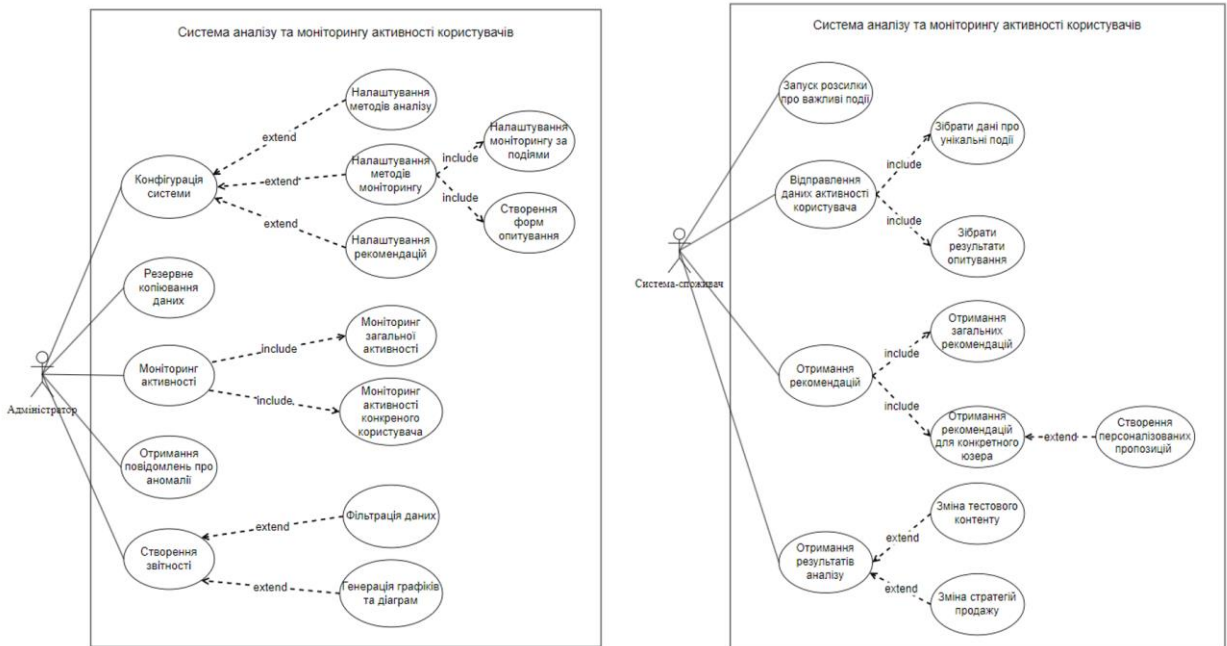
	Cohort Value	0	1	2	3	4	5	6	7	8	9	10	11
Apr 2022	21	100%	100%	95.24%	95%	100%	100%	100%	100%	94.74%	100%	100%	100%
May 2022	30	100%	100%	100%	93.33%	100%	100%	100%	96.43%	96.3%	100%	103.85%	
Jun 2022	30	100%	100%	100%	90%	100%	96.3%	100%	100%	96.15%	104%		
Jul 2022	28	100%	100%	100%	92.86%	100%	100%	100%	100%	100%			
Aug 2022	35	100%	97.14%	100%	85.24%	103.33%	100%	96.77%	96.67%				
Sep 2022	53	100%	100%	100%	92.45%	100%	100%	97.96%					
Oct 2022	44	100%	100%	97.73%	93.02%	95%	97.37%						
Nov 2022	48	97.92%	100%	93.62%	90.91%	92.0%							
Dec 2022	49	100%	93.88%	97.83%	93.11%								
Jan 2023	67	100%	100%	100%									
Feb 2023	62	100%	100%										
Mar 2023	80	100%											
Average	46	99.82%	99.14%	98.5%	91.74%	98.46%	99.08%	98.88%	98.45%	96.94%	101.45%	102.27%	100%

# Специфікація вимог

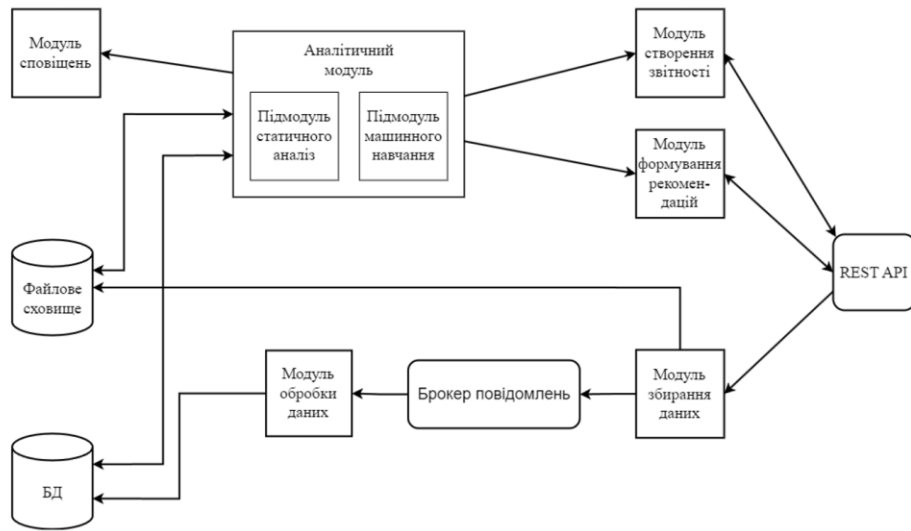
## Існуючим продуктам не вистачає таких відмінних особливостей:

- Здатність до аналізу багатовимірних даних. Системи не мають достатньої потужності для ефективного аналізу багатовимірних даних, включаючи різні джерела інформації, що обмежує можливості виявлення складних взаємозв'язків.
- Адаптивність до особливостей проекту. Система повинна бути гнучкою та легко налаштовуватися для підтримки різних сценаріїв використання та врахування специфічних особливостей програмного продукту.
- Масштабованість та продуктивність. Зі збільшенням числа користувачів та обсягу даних система має залишатися продуктивною. Масштабованість є критичним фактором, щоб забезпечити ефективне функціонування в умовах постійної активності користувачів.
- Інтелектуальний аналіз даних. Існуючі рішення зазвичай надають базовий аналіз користувальницької активності, але їхня здатність до глибокого та інтелектуального аналізу даних залишається обмеженою.

## Діаграми прецедентів



## Структура серверної частини системи UBA

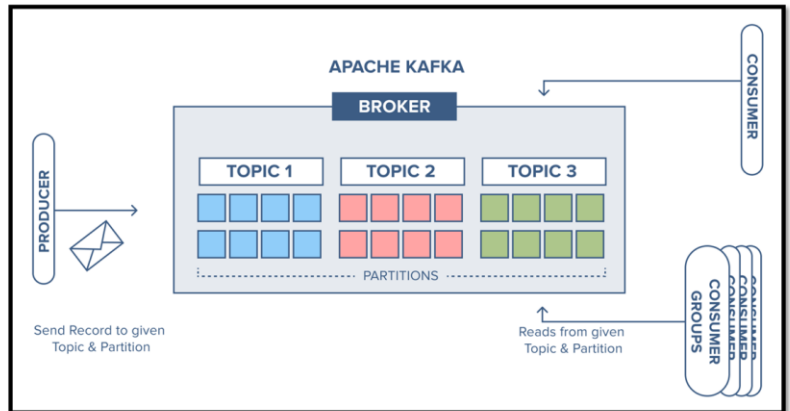


## Ефективна обробка зібраних даних

Основне завдання брокера повідомлень - забезпечувати асинхронний обмін даними між компонентами, дозволяючи їм взаємодіяти незалежно від розташування та часу роботи. Це дозволяє надсилати повідомлення без очікування відповіді, розділити процеси та обробляти їх у фоновому режимі.

При необхідності, брокер може виконувати попередню обробку даних, наприклад фільтрацію.

```
@KafkaListener(groupId = "userClickGroup", topics = "userClicks")
public void consumeUserClicks(@Payload String message, Acknowledgment acknowledgment) {
    //Обробка повідомлення, значення вилучено з тіла Kafka-повідомлення
    userActivityHandler.handle(message);
    //Надсилаємо підтвердження про успішну обробку повідомлення
    acknowledgment.acknowledge();
}
```



## Аналіз даних активності користувачів

### Статичний метод

метод, який використовується для опису даних, отримання загальних відомостей, виявлення закономірностей та тенденцій, а також для перевірки гіпотез. Статичний аналіз даних означає вивчення даних у нерухомому стані, тобто без урахування часу чи послідовності подій.

Використовується для опису даних, щоб отримати уявлення про їх основні характеристики, такі як середнє, медіана, стандартне відхилення та інше.

### Метод машинного навчання

замість явного програмування, де розробники вказують комп'ютеру конкретні кроки виконання завдання, у машинному навчанні використовується підхід, у якому комп'ютер навчається на даних і робить передбачення чи приймає рішення без явного програмування.

- Допомагає автоматично виявляти групи користувачів зі схожою поведінкою
- Здатне створювати моделі, які передбачають майбутні дії користувачів на основі їхньої попередньої поведінки
- Дозволяє створювати персоналізовані рекомендації та контент
- Може автоматично виявляти аномальну чи незвичайну поведінку

## Оптимізація стратегії електронного маркетингу за допомогою машинного навчання

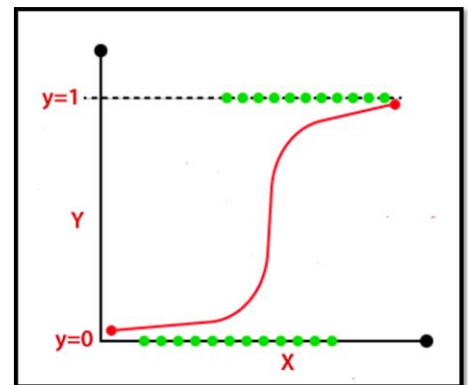
**Що потрібно спрогнозувати** – цільову аудиторію для розсилок електронною поштою.

Будемо модель статистичного зв'язку  $x \rightarrow y$ , де  $Y$  - дії користувача,  $X$  - множина змінних, за допомогою яких передбачається пояснити мінливість змінної  $y$ .

- вік користувача;
- відсоток відкриття листів у будні/вихідні;
- тип повідомлення;
- активність користувача за 5/10/15 днів.
- середня кількість листів одержуваних/відкритих користувачем;

Завдання зводиться до бінарної класифікації. Модель повинна передбачати значення змінної відгуку за заданого значення параметрів  $x_i \in X, i = \{1, \dots, n\}$ .

**Логістична регресія** - метод, основна ідея якого зв'язати лінійну комбінацію вхідних ознак із ймовірністю приналежності до певного класу.



## **Висновки**

- Досліджені сучасні методи моніторингу та аналізу, які дозволяють в повній мірі збирати дані та проводити інтелектуальний аналіз.
- Вибраний та обґрунтований технічний стек для реалізації системи UBA, який забезпечить здатність легко масштабуватись, кросплатформеність та швидку пропускну здатність.
- Розроблено модуль моніторингу, який надає кілька найбільш універсальних способів передачі та збору даних з різних систем.
- Розроблено аналітичний модуль, який використовує статистичні методи, методи машинного навчання для отримання кількісних оцінок та виявлення закономірностей.