**Специфікація програмного забезпечення**

**Вебсистема для рецензій фільмів**

**Software Requirements Specification**

**1.0**

**19.03.2025**

**Решетняк Антон Олексійович**

**Голоха Нікіта Едуардович**

**Рубаненко Марія Сергіївна**

**ІСТОРІЯ ЗМІН**

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Опис | Автор | Коментарі |
| 20.03.2025 | Створено пункти 1.1 - 1.3 | Рубаненко Марія Сергіївна |  |
| 01.04.2025 | Створено пункти 1.4 - 1.5, 2.1 - 2.3 | Голоха Нікіта Едуардович |  |
| 05.04.2025 | Створено пункти 2.4, 2.5, 3.1 - 3.2 | Решетняк Антон Олексійович |  |

**ЗМІСТ**

[1 Вступ 5](#_Toc201936911)

[1.1 Огляд продукту. 5](#_Toc201936912)

[1.2 Мета 6](#_Toc201936913)

[1.3 Межі 7](#_Toc201936914)

[1.4 Посилання. 9](#_Toc201936915)

[1.5 Означення та абревіатури. 11](#_Toc201936916)

[2 Загальний опис 13](#_Toc201936917)

[2.1 Перспективи продукту 13](#_Toc201936918)

[2.2 Функції продукту 13](#_Toc201936919)

[2.3 Характеристики користувачів 13](#_Toc201936920)

[2.4 Загальні обмеження 14](#_Toc201936921)

[2.5 Припущення і залежності 14](#_Toc201936922)

[3 Конкретні Вимоги 15](#_Toc201936923)

[3.1 Вимоги до зовнішніх інтерфейсів 15](#_Toc201936924)

[3.1.1 Інтерфейс користувача. 15](#_Toc201936925)

[3.1.2 Апаратний інтерфейс 16](#_Toc201936926)

[3.1.3 Програмний інтерфейс 17](#_Toc201936927)

[3.1.4 Комунікаційний протокол 19](#_Toc201936928)

[3.1.5 Обмеження пам’яті 21](#_Toc201936929)

[3.1.6 Операції 22](#_Toc201936930)

[3.1.7 Функції продукту. 24](#_Toc201936931)

[3.1.8 Припущення й залежності 26](#_Toc201936932)

[3.2 Властивості програмного продукту 27](#_Toc201936933)

[3.3 Атрибути програмного продукту 29](#_Toc201936934)

[3.3.1 Надійність 29](#_Toc201936935)

[3.3.2 Доступність 30](#_Toc201936936)

[3.3.3 Безпека 30](#_Toc201936937)

[3.3.4 Супроводжуваність 30](#_Toc201936938)

[3.3.5 Переносимість 31](#_Toc201936939)

[3.3.6 Продуктивність 31](#_Toc201936940)

[3.4 Вимоги бази даних. 31](#_Toc201936941)

[3.5 Інші вимоги. 32](#_Toc201936942)

[4 Додаткові матеріали 35](#_Toc201936943)

# **1 ВСТУП**

## 1.1 Огляд продукту.

Сучасні бізнеси, особливо у сфері електронної комерції, щодня накопичують великі обсяги подій користувацької активності. Проте ці дані часто залишаються неструктурованими, розрізненими та малопридатними для системного аналізу. Без централізованої обробки, надійного збереження та автоматизації обчислювальних процесів бізнес втрачає змогу оперативно реагувати на зміни в поведінці користувачів, оцінювати ефективність маркетингових кампаній або приймати обґрунтовані рішення на основі даних.

У відповідь на ці виклики буде створено хмарну аналітичну дата-платформу, яка забезпечить наскрізний шлях обробки даних — від збору raw-подій до формування бізнес-метрик у форматі, придатному для візуалізації. Проєкт фокусується на впровадженні автоматизованого ELT-конвеєра у хмарному середовищі Google Cloud Platform, що дозволить побудувати прозору, масштабовану та адаптивну інфраструктуру для аналітики.

Платформа буде базуватися на сучасній архітектурі медальйонного типу (bronze → silver → gold), у якій:

* bronze-рівень зберігатиме необроблені raw-події;
* silver-рівень формуватиме нормалізовані сутності (сесії, події, транзакції);
* gold-рівень агрегуватиме метрики для аналітичного використання.

Керування даними в цій системі здійснюватиметься за допомогою оркестратора Apache Airflow, а основним інструментом трансформації буде dbt (Data Build Tool). Для побудови інтерфейсів візуалізації буде використано Apache Superset, що забезпечить зручну взаємодію бізнес-користувачів із системою.

У результаті реалізації платформи буде досягнуто:

* автоматизоване перетворення сирих GA4-подій у структуровані датамарти;
* підтримку щоденної побудови метрик із деталізацією за SKU, брендом, країною тощо;
* гнучку фільтрацію та адаптацію дашбордів під запити продуктових, маркетингових та бізнес-аналітиків;
* високу прозорість і повторюваність усіх етапів обробки даних.

## 1.2 Мета

Метою проєкту буде розробка хмарної аналітичної дата-платформи, яка забезпечить повний цикл обробки подій користувачів — від отримання сирих даних до вивантаження агрегованих метрик у BI-інтерфейс. Система буде орієнтована на потреби електронної комерції та дозволить реалізувати прозору, масштабовану й автоматизовану інфраструктуру аналітики на базі хмарних сервісів Google Cloud Platform (GCP).

Платформа буде реалізована за допомогою наступних технологій:

* Apache Airflow як оркестратора для побудови автоматизованих DAG-процесів;
* DLT (Data Loading Tool) для інкрементального завантаження raw-даних із зовнішніх джерел (зокрема GA4, Stripe, Adyen);
* dbt (Data Build Tool) як основного інструмента для розробки SQL- та Python-моделей у шарі трансформацій;
* BigQuery як аналітичного сховища, яке забезпечить швидке виконання запитів і гнучке масштабування;
* Apache Superset для побудови інтерактивних дашбордів і візуалізації бізнес-метрик.

У межах проєкту буде реалізовано:

* автоматизований ELT-конвеєр, який щоденно оброблятиме події у форматі GA4;
* медальйонну архітектуру даних із розмежуванням на bronze, silver та gold рівні;
* аналітичні моделі, що охоплюватимуть ключові бізнес-напрямки: продуктову, маркетингову й комерційну аналітику;
* дашборди в Superset, що підтримуватимуть фільтрацію за країнами, брендами, SKU, типами пристроїв тощо.

Основною метою буде побудова надійного аналітичного середовища, яке забезпечить повторюваність обробки, прозорість трансформацій, гнучкість адаптації до нових бізнес-вимог і зручний доступ до метрик для різних команд.

## 1.3 Межі

Розробка хмарної аналітичної дата-платформи передбачає чітке окреслення її меж функціональності, інфраструктури, зони відповідальності окремих компонентів, а також умов доступу, масштабування та експлуатації. Такий підхід дозволяє уникнути неузгодженостей між компонентами системи, забезпечити правильний розподіл обов’язків і стабільну роботу платформи навіть при зростанні навантаження.

Розгортання системи відбуватиметься виключно у хмарному середовищі Google Cloud Platform (GCP), із дотриманням принципів Infrastructure as Code. Інфраструктура буде створюватися та керуватися за допомогою Terraform або іншого відповідного інструменту IaC. Всі компоненти платформи будуть ізольовані в межах окремого GCP-проєкту з чітко визначеними правами доступу відповідно до ролей. Оркестрація процесів ETL буде реалізована через Apache Airflow, розгорнутий на Cloud Composer або Kubernetes-кластері для забезпечення масштабованості. Всі DAG-файли, dbt-моделі та конфігурації зберігатимуться у централізованому Git-репозиторії, а їхнє розгортання відбуватиметься через CI/CD пайплайни. Доступ до аналітичних дашбордів у Superset буде дозволено лише авторизованим користувачам через інтеграцію з GCP IAM або SSO.

Інфраструктурні та архітектурні межі передбачають обмеження джерел даних — лише ті сервіси, які експортують дані у GCP. Це, зокрема, Google Analytics 4 (через BigQuery Export), платіжні сервіси на кшталт Stripe чи Adyen (через webhooks і їхню подальшу обробку), а також інші події, що можуть логуватися у Cloud Storage у форматі JSON. Архітектура буде поділена на три рівні: на bronze-рівні зберігатимуться сирі події без змін, silver-рівень включатиме нормалізацію, об’єднання, фільтрацію та очищення даних, а gold-рівень — це агреговані таблиці з основними метриками для аналітики. Система не виконуватиме модифікації сирих даних, не оброблятиме зовнішні API-запити та не реалізовуватиме live-аналітику в реальному часі — передбачена щогодинна або щоденна обробка.

Аналітичне сховище буде побудовано виключно на основі Google BigQuery, без інтеграції з іншими OLAP або NoSQL системами. Зберігання даних буде реалізовано у форматі денормалізованих таблиць, що розділятимуться на raw-, silver- та gold-рівні, кожен з яких представлений окремими моделями у dbt. Дані не зберігатимуться довше встановленого терміну (наприклад, 18 місяців для GA4). Доступ до сховища регулюватиметься за допомогою ACL та конфігурацій dbt-permissions. Таблиці, які використовуються для візуалізації, будуть оптимізовані за розміром і структурою завдяки використанню ephemeral-моделей, CTE або view-моделей.

Трансформації та обробка даних будуть реалізовуватися виключно засобами dbt — за допомогою SQL- і Python-моделей. Система не включатиме повноцінну бізнес-логіку або ML-рішення, однак зосередиться на відтворюваних трансформаціях (reproducible transformations), які можна повністю оновлювати. Усі складні обчислення (наприклад, retention, conversion, revenue per session) будуть реалізовані на рівні gold.

Візуалізація даних буде обмежена інтерфейсом Apache Superset без інтеграції з іншими BI-інструментами. Платформа не буде автоматично генерувати звіти або презентації, надсилати сповіщення чи дозволяти завантаження повних дампів даних через інтерфейс. Superset працюватиме виключно з попередньо агрегованими моделями з gold-шару, підтримуватиме фільтри на рівні дашбордів, блоки KPI, часові ряди, таблиці, воронки, теплокарти та карти.

Такий підхід до визначення меж дозволяє забезпечити чітку структуру, масштабованість, контроль доступу і високу якість аналітики в рамках розгортання дата-платформи.

## 1.4 Посилання

У цій специфікації згадуються такі нормативні документи, стандарти та технічні ресурси:

1. Adobe experience platform documentation. Experience League | Adobe. URL: <https://experienceleague.adobe.com/en/docs/experience-platform> (дата звернення: 20.03.2025).

2. Marketing cloud | salesforce developer center. Developer Portal | Salesforce Developers. URL: <https://developer.salesforce.com/developer-centers/marketing-cloud> (дата звернення: 21.03.2025).

3. Google analytics for developers | google for developers. Google for Developers. URL: <https://developers.google.com/analytics/devguides/collection/ga4?hl=en> (дата звернення: 20.03.2025).

4. Build a dlt pipeline | dlt Docs. dltHub: ELT as Python Code. URL: <https://dlthub.com/docs/tutorial/load-data-from-an-api> (дата звернення: 25.03.2025).

5. Quick start – airflow documentation. Apache Airflow. URL: <https://airflow.apache.org/docs/apache-airflow/2.10.5/start.html> (дата звернення: 28.03.2025).

6. Cloud composer overview | google cloud. Google Cloud. URL: <https://cloud.google.com/composer/docs/composer-3/composer-overview> (дата звернення: 28.03.2025).

7. BigQuery overview | google cloud. Google Cloud. URL: <https://cloud.google.com/bigquery/docs/introduction> (дата звернення: 29.03.2025).

8. BigQuery setup | dbt Developer Hub. dbt Developer Hub. URL: <https://docs.getdbt.com/docs/core/connect-data-platform/bigquery-setup> (дата звернення: 30.03.2025).

9. DAGs – airflow documentation. Apache Airflow. URL: <https://airflow.apache.org/docs/apache-airflow/2.10.5/core-concepts/dags.html> (дата звернення: 01.04.2025).

10. Intro | superset. Welcome | Superset. URL: <https://superset.apache.org/docs/intro/> (дата звернення: 31.03.2025).

11. What is a medallion architecture?. Databricks. URL: <https://www.databricks.com/glossary/medallion-architecture> (дата звернення: 30.03.2025).

12. Google cloud overview | overview. Google Cloud. URL: <https://cloud.google.com/docs/overview> (дата звернення: 20.03.2025).

13. Airflow.sensors.external\_task\_sensor – airflow documentation. Apache Airflow. URL: <https://airflow.apache.org/docs/apache-airflow/1.10.8/_api/airflow/sensors/external_task_sensor/index.html> (дата звернення: 06.04.2025).

14. What is astronomer cosmos? – astronomer cosmos documentation. URL: <https://astronomer.github.io/astronomer-cosmos/> (дата звернення: 08.04.2025).

15. Testing behavior – astronomer cosmos documentation. URL: <https://astronomer.github.io/astronomer-cosmos/configuration/testing-behavior.html> (дата звернення: 11.04.2025).

16. Airflow trigger rules | Astronomer Documentation. Astronomer: The Best Place to Run Apache Airflow®. URL: <https://www.astronomer.io/docs/learn/airflow-trigger-rules/> (дата звернення: 02.05.2025).

17. Orchestrate dbt core projects with airflow and cosmos | astronomer documentation. Astronomer: The Best Place to Run Apache Airflow®. URL: <https://www.astronomer.io/docs/learn/airflow-dbt/> (дата звернення: 01.05.2025).

18. Welcome to pydantic - pydantic. Welcome to Pydantic - Pydantic. URL: <https://docs.pydantic.dev/latest/> (дата звернення: 13.05.2025).

19. Big Data: Principles and best practices of scalable real-time data systems. Shelter Island, NY : Manning, 2015. 308 p. (дата звернення: 13.03.2025)

## 1.5 Означення та абревіатури

У цьому документі використовуються наступні терміни та скорочення:

CRM (Customer Relationship Management) — система управління взаємовідносинами з клієнтами, що використовується для зберігання, обробки та аналізу контактних даних і комунікацій.

SMS (Short Message Service) — сервіс коротких текстових повідомлень.

GA4 (Google Analytics 4) — платформа аналітики від Google, що дозволяє відстежувати взаємодію користувачів із веб- і мобільними додатками.

SLA (Service Level Agreement) — угода про рівень надання послуг між постачальником і користувачем.

dlt (data load tool) — інструмент для завантаження, очищення та передачі потокових або батчевих даних.

API (Application Programming Interface) — інтерфейс для взаємодії між програмними компонентами.

ELT (Extract Load Transform) — підхід до обробки даних, при якому спочатку дані завантажуються у сховище, а потім трансформуються.

ETL (Extract Transform Load) — класична схема обробки даних із попередньою трансформацією перед завантаженням у сховище.

dbt (data build tool) — інструмент для аналітичної інженерії, що дозволяє будувати трансформаційний шар даних за допомогою SQL та конфігурацій у YAML.

DAG (Directed Acyclic Graph) — орієнтований ациклічний граф, який описує послідовність виконання задач (наприклад, у Apache Airflow).

SQL (Structured Query Language) — мова запитів до реляційних баз даних.

UI (User Interface) — користувацький інтерфейс.

PR (Pull Request) — запит на злиття змін у репозиторій, що дозволяє проводити код-рев’ю.

CI/CD (Continuous Integration / Continuous Delivery) — практика безперервної інтеграції та доставки змін у програмному забезпеченні.

GCP (Google Cloud Platform) — хмарна платформа від Google для обчислень, зберігання, аналізу та інших сервісів.

DB (Database) — база даних.

REST-API (Representational State Transfer Application Programming Interface) — архітектурний стиль створення API, що використовує HTTP-запити.

СУБД (Система управління базами даних) — програмне забезпечення для створення, підтримки та керування базами даних.

HTTP (Hypertext Transfer Protocol) — протокол передачі гіпертексту в інтернеті.

CPU (Central Processing Unit) — центральний процесор.

IAM (Identity and Access Management) — система керування ідентифікацією користувачів і їх доступом до ресурсів.

# **2 ЗАГАЛЬНИЙ ОПИС**

## 2.1 Перспективи продукту

Хмарна аналітична дата-платформа надає бізнес-користувачам (аналітикам, маркетологам) комплексний інструментарій для взаємодії з корпоративними даними та отримання бізнес-інсайтів. Метою проєкту є створення надійної, автоматизованої та масштабованої інфраструктури для збору, обробки, трансформації та візуалізації великих обсягів даних електронної комерції. Розроблений продукт демонструє ефективну взаємодію між джерелами даних, потужним хмарним сховищем BigQuery та інструментами оркестрації і візуалізації. Система має значний потенціал для розвитку, включаючи розширення джерел даних, покращення моделей трансформації, інтеграцію з іншими бізнес-системами та додавання нових аналітичних можливостей.

## 2.2 Функції продукту

Дата-платформа забезпечує автоматизований інжест сирих даних подій користувачів з джерел веб-аналітики у BigQuery. Вона виконує трансформації даних за принципами медальйонної архітектури (Bronze, Silver, Gold), перетворюючи сирі події на очищені бізнес-сутності та агреговані метрики. Користувачі можуть отримувати доступ до готових аналітичних моделей через Apache Superset для побудови інтерактивних дашбордів та звітів. Платформа забезпечує високу якість та достовірність даних.

## 2.3 Характеристики користувачів

Користувачами є бізнес-аналітики, маркетологи, продуктові менеджери та інші фахівці, які зацікавлені в отриманні даних для прийняття рішень та оптимізації бізнес-процесів в електронній комерції. Необхідні базові навички роботи з інструментами бізнес-аналітики (наприклад, Superset) та розуміння бізнес-метрик. У системі є 2 типи користувачів:

* Інженери даних / Розробники: Відповідальні за розробку, підтримку та розширення пайплайнів інжесту та трансформації даних, керування інфраструктурою через Terraform та моніторинг системи. Мають доступ до Cloud Composer, BigQuery, GitHub, Cloud Monitoring.
* Бізнес-користувачі (Аналітики / Маркетологи): Використовують готові дані для побудови звітів та дашбордів, аналізу бізнес-показників, пошуку інсайтів. Взаємодіють переважно з Apache Superset та BigQuery (Gold-рівень).

## 2.4 Загальні обмеження

Доступ до платформи та її компонентів здійснюється через веб-інтерфейси (Cloud Composer, BigQuery, Superset). Для використання необхідне стабільне підключення до Інтернету та сучасний веб-браузер. Продуктивність та доступність зовнішніх джерел даних можуть впливати на час вивантаження даних. Витрати на хмарні ресурси GCP залежать від обсягів даних та інтенсивності використання.

## 2.5 Припущення і залежності

Припускається, що основними джерелами даних є веб-аналітика та інші системи електронної комерції. Дата-платформа залежить від доступності сервісів Google Cloud Platform (BigQuery, Cloud Composer, Cloud Storage, Compute Engine, Cloud SQL) для зберігання даних, оркестрації та візуалізації. Користувачі Superset мають бути забезпечені необхідними дозволами IAM для доступу до BigQuery. Ефективність трансформацій dbt залежить від коректності та оптимізації SQL-моделей.

# **3 КОНКРЕТНІ ВИМОГИ**

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача.

Інтерфейс користувача аналітичної платформи буде реалізовано у вигляді вебзастосунку з інтерактивними дашбордами, створеними в Apache Superset. Основною метою є забезпечення швидкого та зручного доступу до аналітичної інформації для ключових ролей — маркетологів, продуктових аналітиків і менеджерів з продажу. Всі візуальні компоненти платформи будуть адаптовані до потреб прийняття рішень, спираючись на агреговані метрики та деталізовані зрізи даних.

Побудова інтерфейсу базується на принципі ієрархічної глибини: користувач спочатку отримує загальне уявлення про стан справ за допомогою ключових показників ефективності (KPI), а далі може перейти до вивчення часових трендів, категоріальних порівнянь і детальних таблиць. У верхній частині кожного дашборду буде розміщено короткий текстовий опис його призначення, що допомагає зорієнтуватися в аналітичному контексті. Графіки будуть згруповані за змістом у логічні блоки, кожен із яких позначено заголовком або маркуванням у markdown-блоці, що сприятиме структурованому сприйняттю інформації.

Особливу увагу приділено системі фільтрації: користувачі зможуть динамічно перемикатися між країнами, часовими періодами, категоріями товарів, брендами, SKU та типами пристроїв. Фільтри будуть реалізовані на рівні UI Superset без потреби писати SQL-запити. Завдяки використанню gold-моделей з попередньо агрегованими даними взаємодія з фільтрами буде швидкою й зручною навіть при великих обсягах інформації.

Інтерфейс дашбордів також буде витриманий у єдиному стилі з дотриманням принципів консистентного UI-дизайну: єдина палітра кольорів, чітка ієрархія шрифтів, узгоджена структура розташування елементів. Це зробить взаємодію з платформою інтуїтивною та приємною навіть для користувачів без глибокої технічної підготовки.

Таким чином, інтерфейс користувача виконуватиме не лише роль засобу візуалізації даних, а й стане повноцінним інструментом бізнес-аналітики, що дозволяє швидко реагувати на зміни в поведінці користувачів, ефективності маркетингових кампаній і результативності продажів.

### 3.1.2 Апаратний інтерфейс

У контексті хмарної аналітичної дата-платформи, прямий апаратний інтерфейс до фізичних компонентів відсутній. Програмна система взаємодіє з віртуалізованими апаратними ресурсами, що надаються Google Cloud Platform (GCP). Вимоги до апаратного інтерфейсу визначаються необхідною продуктивністю, масштабованістю та надійністю цих віртуалізованих ресурсів для забезпечення функціонування всіх компонентів платформи.

1. Обчислювальні ресурси (Compute Resources):
   1. Процесор (CPU): Повинні забезпечувати достатню обчислювальну потужність для виконання завдань трансформації даних (dbt), оркестрації (Apache Airflow), роботи BI-інструменту (Apache Superset) та інших фонових процесів. Забезпечується віртуальними ядрами (vCPU) машин GCP.
   2. Оперативна пам'ять (RAM): Достатній обсяг оперативної пам'яті для ефективного кешування даних та виконання ресурсоємних операцій з даними, зокрема для Apache Airflow (Cloud Composer) та Apache Superset (Compute Engine).
2. Ресурси зберігання даних (Storage Resources):
   1. Пропускна здатність диска (Disk I/O): Висока пропускна здатність для операцій читання/запису даних, особливо для Cloud Storage (для сирих даних та DAGs) та BigQuery (для ефективного завантаження та запитів).
   2. Ємність зберігання: Масштабована ємність для зберігання зростаючих обсягів даних у BigQuery, Cloud Storage та Cloud SQL.
3. Мережеві ресурси (Network Resources):
   1. Пропускна здатність мережі (Network Bandwidth): Висока пропускна здатність між хмарними сервісами (наприклад, між Cloud Storage та BigQuery, між Airflow та BigQuery), а також для доступу до зовнішніх джерел даних та BI-інструменту користувачами.
   2. Затримка (Latency): Низька затримка мережі між компонентами системи для забезпечення оперативної обробки даних та формування звітів.
4. Спеціалізовані ресурси (якщо потрібно):
   1. На поточному етапі розвитку, використання графічних процесорів (GPU) або тензорних процесорів (TPU) не є обов'язковим. Однак, архітектура повинна мати потенційну можливість інтеграції таких ресурсів у майбутньому для завдань машинного навчання чи складних аналітичних моделей.

Всі перераховані апаратні ресурси надаються та управляються Google Cloud Platform, що знімає необхідність прямого адміністрування фізичного обладнання та забезпечує гнучке масштабування відповідно до потреб системи.

### 3.1.3 Програмний інтерфейс

Програмний інтерфейс (API) визначає способи взаємодії розробленої системи з іншими програмними компонентами, зовнішніми джерелами даних та системами-споживачами. Для хмарної аналітичної дата-платформи він охоплює наступні аспекти:

1. інтерфейси для збору даних (Ingestion Interfaces):
   1. RESTful/SOAP API: Для програмного підключення до зовнішніх SaaS-додатків, веб-сервісів та інших онлайн-джерел даних (наприклад, через dlt);
   2. SQL-інтерфейси баз даних: Для підключення до реляційних баз даних (наприклад, PostgreSQL, MySQL) з метою вивантаження даних (через dlt);
   3. файлові інтерфейси (Cloud Storage API): Для імпорту даних у форматах CSV, JSON, Parquet, Avro з об'єктних сховищ, таких як Google Cloud Storage;
   4. інтерфейси спеціалізованих коннекторів: Використання готових коннекторів dlt для взаємодії з популярними системами (наприклад, GA4, CRM, ERP), що абстрагують специфіку їхніх API;
2. внутрішні інтерфейси компонентів платформи:
   1. BigQuery Standard SQL API: Основний інтерфейс для виконання запитів та трансформацій даних за допомогою dbt, а також для отримання даних інструментами BI;
   2. Apache Airflow API/UI (Cloud Composer): Програмний та користувацький інтерфейс для управління DAG-ами, моніторингу їх виконання та взаємодії з оркестратором;
   3. Google Cloud Storage API: Інтерфейс для зберігання та доступу до сирих даних, файлів DAGs та інших артефактів;
   4. Cloud SQL API: Інтерфейс для взаємодії з керованою базою даних для зберігання метаданих Apache Superset;
3. інтерфейси для споживання та візуалізації даних (Consumption & Visualization Interfaces):
   1. SQL-інтерфейс (Apache Superset до BigQuery): Інтерфейс, який дозволяє Apache Superset підключатися до BigQuery (зокрема до шару Gold) та виконувати запити для побудови дашбордів та візуалізацій;
   2. укспортні інтерфейси: Можливість експорту оброблених даних з BigQuery у різні формати (CSV, JSON, Parquet) до Cloud Storage для подальшого використання іншими системами або додатками;
4. інтерфейси управління та розгортання (Management & Deployment Interfaces):
   1. Git (GitHub API): Інтерфейс для керування версіями коду (DAGs, dbt-моделі, конфігурації Terraform), спільної роботи та інтеграції з CI/CD;
   2. Terraform CLI/API: Програмний інтерфейс для декларативного управління та розгортання інфраструктури GCP як коду;
   3. Docker CLI/API: Інтерфейс для управління контейнерами, які використовуються для розгортання окремих компонентів (наприклад, Superset);
   4. Google Cloud Console API: Дозволяє програмно взаємодіяти з різними сервісами GCP для автоматизації управління та моніторингу.

### 3.1.4 Комунікаційний протокол

Комунікаційні протоколи визначають правила та формати обміну даними між компонентами програмної системи, а також із зовнішніми сервісами. У контексті хмарної аналітичної дата-платформи використовуються наступні протоколи:

1. HTTPS (Hypertext Transfer Protocol Secure):
   1. призначення: Основний протокол для безпечної передачі даних через Інтернет. Використовується для доступу до веб-інтерфейсів (наприклад, Apache Superset UI, GCP Console), взаємодії з API зовнішніх джерел даних (через dlt), а також для внутрішньої комунікації між сервісами GCP;
   2. особливості: Забезпечує шифрування даних (TLS/SSL) та автентифікацію серверів, гарантуючи конфіденційність та цілісність переданої інформації;
2. TCP/IP (Transmission Control Protocol/Internet Protocol):
   1. призначення: Базовий набір протоколів для комунікації в комп'ютерних мережах, включаючи Інтернет. Всі мережеві взаємодії на рівні інфраструктури GCP (між BigQuery, Cloud Composer, Cloud Storage, Compute Engine) базуються на TCP/IP;
   2. особливості: Забезпечує надійну, впорядковану та з перевіркою помилок доставку пакетів даних;
3. SQL (Structured Query Language):
   1. призначення: Мова запитів для взаємодії з реляційними базами даних та аналітичними сховищами даних. Використовується dbt для виконання трансформацій у BigQuery та Apache Superset для запиту даних з BigQuery;
   2. особливості: Стандартизована мова, що дозволяє отримувати, маніпулювати та визначати дані у BigQuery та Cloud SQL;
4. протоколи API (специфічні для сервісів):
   1. Google Cloud APIs: Різноманітні протоколи та формати (наприклад, REST, gRPC) для програмної взаємодії між компонентами платформи та сервісами GCP (BigQuery API, Cloud Storage API, Cloud Composer API, IAM API);
   2. dlt Source APIs: Конкретні протоколи (залежно від джерела даних), які використовуються dlt для вилучення даних з різних сторонніх систем (наприклад, OAuth2 для автентифікації до деяких веб-сервісів);
5. Git Protocol (SSH/HTTPS):
   1. призначення: Використовується для взаємодії між локальним середовищем розробки (Git) та віддаленим репозиторієм GitHub для операцій версійного контролю (push, pull, clone);
   2. особливості: Забезпечує безпечну та ефективну синхронізацію кодової бази проєкту.

### 3.1.5 Обмеження пам’яті

У хмарній аналітичній дата-платформі, розгорнутій на Google Cloud Platform, "обмеження пам'яті" не є фіксованими фізичними лімітами, як у традиційних локальних системах. Натомість, вони стосуються вимог до доступності та ефективного використання віртуалізованих ресурсів оперативної пам'яті, що динамічно надаються хмарним провайдером. Ці обмеження визначаються балансом між необхідною продуктивністю, масштабованістю та оптимізацією витрат.

Google BigQuery: Як бессерверне сховище даних, BigQuery абстрагує безпосереднє управління пам'яттю. Його продуктивність для обробки запитів і трансформацій залежить від доступних слотів обробки, які динамічно масштабуються Google. Відповідно, вимоги до "пам'яті" тут стосуються здатності BigQuery ефективно оперувати великими обсягами даних без прив'язки до фіксованих обсягів ОЗП.

Apache Airflow (Cloud Composer): Робочі вузли (workers) та планувальник (scheduler) Cloud Composer вимагають достатнього обсягу оперативної пам'яті для ефективного виконання завдань DAGs, особливо тих, що передбачають обробку даних у пам'яті або виконання складних обчислень. Обмеження пам'яті для цих компонентів визначаються обраним типом машини та конфігурацією Cloud Composer, впливаючи на паралелізм виконання завдань та стійкість до збоїв.

Apache Superset (Compute Engine): Віртуальна машина (VM) Google Compute Engine, на якій розгорнуто Apache Superset, повинна мати достатній обсяг оперативної пам'яті для забезпечення стабільної роботи інтерфейсу, швидкого завантаження дашбордів та обробки запитів від користувачів. Недостатній обсяг пам'яті призведе до зниження продуктивності інтерфейсу та потенційних збоїв при великому навантаженні або складних візуалізаціях.

Процеси інжестування та трансформації (dlt, dbt): Хоча dbt виконує трансформації безпосередньо у BigQuery (що мінімізує локальні потреби в пам'яті), інструмент dlt може вимагати певний обсяг пам'яті на етапі вилучення та завантаження даних, залежно від розміру порцій даних, що обробляються, та складності схем.

Загальним обмеженням є економічна доцільність та оптимізація ресурсів. Хоча GCP дозволяє масштабувати пам'ять практично без обмежень, надмірне виділення ресурсів, що не використовуються, призведе до необґрунтованих витрат. Таким чином, вибір конфігурації пам'яті базується на компромісі між забезпеченням необхідної продуктивності та контролем операційних витрат.

### 3.1.6 Операції

1. розгортання та оновлення (Deployment & Updates):
   1. розгортання всіх компонентів платформи та оновлення інфраструктури повинно здійснюватися автоматизовано за принципами Infrastructure as Code (IaC) за допомогою Terraform;
   2. оновлення та синхронізація DAG-ів Apache Airflow, а також інших компонентів, що розгортаються, повинні виконуватися автоматично через CI/CD пайплайни GitHub Actions;
   3. система повинна підтримувати версійність всіх конфігурацій та коду через Git для контрольованого відкату змін у разі потреби;
2. моніторинг та логування (Monitoring & Logging):
   1. усі компоненти системи (Cloud Composer, BigQuery, Compute Engine, Cloud Storage) повинні генерувати логи, доступні для централізованого збору та аналізу через Google Cloud Logging;
   2. необхідно налаштувати моніторинг ключових метрик продуктивності (навантаження CPU, використання пам'яті, IOPS, латентність запитів) та стану системи за допомогою Google Cloud Monitoring;
   3. для моніторингу виконання робочих процесів та завдань ETL/ELT слід використовувати веб-інтерфейс Apache Airflow (Cloud Composer UI) ;
   4. моніторинг якості даних повинен здійснюватися через результати тестів dbt та кастомні перевірки, інтегровані в пайплайни;
3. сповіщення (Alerting):
   1. система повинна автоматично генерувати сповіщення (наприклад, через Google Cloud Monitoring Alerts) у разі виникнення критичних помилок, аномалій у виконанні завдань, перевищення порогових значень продуктивності або проблем з якістю даних;
   2. сповіщення повинні надходити відповідним командам для оперативного реагування;
4. обслуговування та підтримка (Maintenance & Support):
   1. для керованих сервісів GCP (BigQuery, Cloud Composer, Cloud SQL) оновлення та виправлення помилок забезпечуються провайдером;
   2. необхідно передбачити періодичне оновлення версій компонентів, які розгортаються користувачем (наприклад, Apache Superset), згідно з планом підтримки та безпеки;
   3. політики зберігання даних у BigQuery та Cloud Storage повинні бути чітко визначені та автоматизовані;
5. відновлення після збоїв та резервне копіювання (Disaster Recovery & Backup):
   1. система повинна використовувати вбудовані механізми резервного копіювання та відновлення, що надаються GCP для BigQuery, Cloud Storage та Cloud SQL;
   2. пайплайни даних повинні бути ідемпотентними, щоб дозволяти повторний запуск завдань без дублювання або пошкодження даних;
6. управління безпекою (Security Operations):
   1. постійний моніторинг доступу та активності користувачів через Google Cloud IAM та Cloud Audit Logs;
   2. регулярний перегляд та оновлення політик безпеки та прав доступу;
7. управління масштабованістю (Scalability Management):
   1. система повинна мати можливість динамічного масштабування ресурсів GCP (BigQuery, Cloud Composer, Compute Engine) відповідно до зростаючих потреб у обробці даних та навантаженні.

### 3.1.7 Функції продукту.

Розроблювана хмарна аналітична дата-платформа забезпечуватиме повний цикл роботи з даними електронної комерції — від збору подій із зовнішніх джерел до побудови аналітичних моделей і візуалізацій. Основна мета — створити інструмент, який дозволить компаніям ухвалювати обґрунтовані data-driven рішення завдяки системній та автоматизованій роботі з аналітичними даними.

Платформа охоплює шість ключових функціональних напрямів. Перший — збір та завантаження даних. Система автоматично отримуватиме інформацію з зовнішніх джерел, таких як Google Analytics 4 чи Stripe, використовуючи API або webhook. Дані надходитимуть у хмарне сховище Google Cloud Storage або безпосередньо в Google BigQuery, організовуватимуться у вигляді сирих подій на bronze-шарі з логічним поділом за джерелами.

Другим етапом є трансформація та збагачення даних, що здійснюється через dbt за сучасною практикою ELT. На silver-рівні формуватимуться нормалізовані таблиці за ключовими бізнес-сутностями — сесіями, транзакціями, товарами, подіями взаємодії, користувачами. На gold-рівні дані агрегуватимуться для розрахунку метрик, таких як кількість сесій, доходи, конверсії, середній час взаємодії тощо. З метою ефективності використовуються ефемерні моделі та CTE, що дозволяє повторно застосовувати обчислення без дублювання коду.

Третій блок — оркестрація та автоматизація, реалізована за допомогою Apache Airflow. DAG-пайплайни оновлюють моделі поетапно, відповідно до логіки переходу між bronze, silver і gold шарами. Система веде журнал обробки, контролює залежності між задачами і автоматично повідомляє про помилки.

Четверта група функцій стосується аналітичної моделі та сегментації. Платформа підтримує сегментний аналіз на різних рівнях: від SKU і категорій товарів до типів пристроїв та країн. Вона дозволяє розраховувати широкий спектр бізнес-метрик — від коефіцієнтів конверсії до ефективності рекламних кампаній. Платформа також підтримує побудову часових моделей (retention), аналіз воронок і оцінку промоакцій, з перспективою впровадження ML-моделей у майбутньому.

Наступний блок — візуалізація та доступ до даних. Платформа інтегрується з Apache Superset, що дозволяє створювати інтерактивні дашборди на основі агрегованих даних із gold-рівня. Доступні засоби фільтрації за країнами, категоріями, брендами, часовими періодами — без потреби у написанні SQL. Дані візуалізуються у вигляді графіків, теплових карт, мап, таблиць, воронок. Передбачено реалізацію трьох ключових дашбордів: Marketing Funnel Report, Product Performance Overview та SKU-level Performance Insights, кожен із яких орієнтований на окремі бізнес-ролі.

Останній функціональний напрям — гнучкість, масштабованість і підтримка. Платформа побудована на серверлес-архітектурі BigQuery, що дозволяє масштабувати обчислення без втручання в інфраструктуру. Центральне управління моделями здійснюється через dbt, а всі зміни проходять через CI/CD пайплайни. Система дотримується високих стандартів якості даних, включно з автоматичним тестуванням, валідацією та доступом, який регулюється за допомогою IAM-політик Google Cloud.

Загалом, функціональність платформи виходить за межі звичайної звітності — вона формує цілісну аналітичну екосистему, що дозволяє компаніям швидко реагувати на зміни в поведінці користувачів, оптимізувати маркетинг, підвищувати ефективність продуктової лінійки та приймати обґрунтовані стратегічні рішення.

### 3.1.8 Припущення й залежності

Припущення (Assumptions):

* + доступність сервісів GCP: Припускається стабільна та безперебійна робота всіх основних сервісів Google Cloud Platform (BigQuery, Cloud Composer, Cloud Storage, Compute Engine, IAM) згідно з угодами про рівень обслуговування (SLA) від Google;
  + стабільність зовнішніх джерел даних: Припускається стабільна доступність та передбачувана структура даних від зовнішніх джерел (наприклад, GA4 API), з яких здійснюється інжестування даних. Зміни в API або схемі даних зовнішніх джерел можуть вимагати оновлення конвеєрів;
  + наявність необхідних облікових даних та дозволів: Припускається наявність та коректна конфігурація всіх необхідних облікових даних, сервісних акаунтів та дозволів (IAM) для безперешкодної взаємодії між компонентами системи та зовнішніми сервісами;
  + достатній бюджет GCP: Припускається наявність достатнього бюджету Google Cloud для покриття витрат на використання всіх сервісів, масштабування ресурсів та зберігання даних відповідно до потреб проєкту;
  + зміни в обсягах даних: Хоча платформа масштабована, припускається, що зростання обсягів даних буде відбуватися у прогнозованих межах, не вимагаючи кардинальної зміни архітектури;
  + зміни в інструментах з відкритим кодом: Припускається, що обрані інструменти з відкритим кодом (dbt, dlt, Apache Superset, Apache Airflow) будуть підтримуватися спільнотою та не матимуть критичних змін, що впливають на сумісність або функціональність.

Залежності (Dependencies):

* + залежність від Google Cloud Platform: Система повністю залежить від інфраструктури та сервісів GCP. Відмова будь-якого критичного сервісу GCP може призвести до збоїв у роботі платформи;
  + залежність від зовнішніх джерел даних: Безперервна робота конвеєрів ELT залежить від доступності та коректності даних, що надходять від зовнішніх систем (наприклад, GA4) ;
  + залежність від GitHub: Для версійного контролю коду та автоматизованих процесів CI/CD (GitHub Actions) система повністю залежить від доступності та функціональності GitHub;
  + залежність від dbt та dlt: Коректність трансформацій даних та ефективність інжестування повністю залежать від стабільної роботи, оновлень та функціональності інструментів dbt та dlt;
  + залежність від Docker: Компоненти, які розгортаються у контейнерах (наприклад, Superset), залежать від коректної роботи Docker-середовища;
  + залежність від Terraform: Управління інфраструктурою та її оновлення залежать від коректного функціонування Terraform.

## 3.2 Властивості програмного продукту

1. масштабованість (Scalability):
   1. програмний продукт спроєктовано з урахуванням горизонтальної масштабованості, що дозволяє обробляти зростаючі обсяги даних та збільшувати кількість користувачів без значної деградації продуктивності;
   2. використання керованих сервісів GCP, таких як BigQuery та Cloud Composer (Apache Airflow), забезпечує автоматичне масштабування обчислювальних ресурсів та сховища відповідно до навантаження;
2. відмовостійкість та надійність (Fault Tolerance & Reliability):
   1. архітектура платформи є відмовостійкою завдяки використанню розподілених та керованих сервісів GCP, які мають вбудовані механізми реплікації даних, автоматичного відновлення та високої доступності;
   2. процеси ELT спроєктовані з урахуванням ідемпотентності, що дозволяє повторний запуск завдань без негативних наслідків для цілісності даних;
   3. регулярне тестування даних за допомогою dbt підвищує надійність аналітичних моделей;
3. автоматизація (Automation):
   1. процеси збору, трансформації та завантаження даних повністю автоматизовані за допомогою Apache Airflow (Cloud Composer);
   2. розгортання та керування інфраструктурою здійснюється автоматично через Terraform (IaC);
   3. процеси безперервної інтеграції та розгортання (CI/CD) реалізовані за допомогою GitHub Actions, що автоматизує оновлення коду та інфраструктури;
4. гнучкість та адаптивність (Flexibility & Adaptability):
   1. модульна архітектура дозволяє легко додавати нові джерела даних, розширювати аналітичні моделі та інтегрувати нові інструменти;
   2. використання dbt забезпечує гнучкість у створенні та модифікації трансформацій даних за допомогою SQL;
   3. платформа дозволяє адаптувати метрики та моделі даних під специфічні бізнес-потреби, що є перевагою над багатьма існуючими аналогами;
5. керованість та моніторинг (Manageability & Monitoring):
   1. система забезпечує централізований моніторинг продуктивності та стану компонентів за допомогою Google Cloud Monitoring та Cloud Logging;
   2. інтерфейси Apache Airflow UI та GCP Console надають засоби для контролю та управління операціями;
6. безпека (Security):
   1. доступ до даних та ресурсів контролюється через Google Cloud IAM (Identity and Access Management), що дозволяє застосовувати деталізовані політики доступу;
   2. дані шифруються як під час передачі (TLS/SSL), так і під час зберігання у BigQuery та Cloud Storage;
7. вартісна ефективність (Cost-effectiveness):
   1. використання хмарних сервісів з оплатою за споживання та оптимізованим масштабуванням дозволяє контролювати операційні витрати, уникаючи значних початкових інвестицій в апаратне забезпечення.

## 3.3 Атрибути програмного продукту

### 3.3.1 Надійність

Усі збої та помилки в роботі пайплайнів відстежуються через систему логування Cloud Composer, що дозволяє оперативно реагувати. Час недоступності критичних сервісів, таких як Cloud Composer та BigQuery, мінімізовано завдяки архітектурі GCP та SLA. Втрата даних у сховищі BigQuery неприпустима завдяки його вбудованим механізмам реплікації та захисту даних. Платформа має автоматичні механізми повторного запуску завдань (retry) у Cloud Composer та можливість відновлення пайплайнів після збоїв.

### 3.3.2 Доступність

Платформа забезпечує високу доступність завдяки використанню керованих сервісів GCP (BigQuery, Cloud Composer, Cloud SQL) з гарантованими SLA. Веб-інтерфейс Apache Superset, розгорнутий на Compute Engine, забезпечує безперебійний доступ до аналітичних дашбордів. Час відгуку на запити до BigQuery оптимізовано завдяки структурі даних та використанню Gold-рівня для аналітики.

### 3.3.3 Безпека

Доступ до ресурсів GCP керується через Google Cloud IAM (Identity and Access Management), забезпечуючи мінімальні привілеї для сервісних акаунтів, визначені у Terraform. Аутентифікація користувачів Apache Superset відбувається через власні механізми Superset, інтегровані з Cloud SQL для зберігання метаданих. Усі взаємодії між компонентами (наприклад, Airflow-BigQuery, Superset-BigQuery) використовують захищені канали зв'язку та механізми авторизації GCP. Вся інфраструктура, визначена Terraform, відповідає політикам безпеки.

### 3.3.4 Супроводжуваність

Код усіх компонентів структурований згідно з принципами модульності та зберігається у Git-репозиторії. Система логування Cloud Composer забезпечує централізоване відстеження виконання завдань, помилок та моніторинг продуктивності. Розгортання та оновлення керуються через GitHub Actions (CI/CD).

### 3.3.5 Переносимість

Центральна логіка пайплайнів базується на мовах Python та SQL, що забезпечує високу гнучкість. Інфраструктура, визначена Terraform, може бути відтворена в будь-якому регіоні GCP або адаптована для інших хмарних провайдерів з підтримкою аналогічних сервісів. Використання Docker-контейнерів для Superset (на VM Instance) та Airflow Workers (у Composer) спрощує розгортання та масштабування.

### 3.3.6 Продуктивність

Система спроможна обробляти значні обсяги даних завдяки масштабованості Google BigQuery та Cloud Composer. Паралельне виконання завдань у Airflow та ефективна робота dbt-моделей забезпечують високу швидкість трансформацій. Час завантаження аналітичних дашбордів у Superset оптимізовано завдяки попередньо агрегованим даним у Gold-рівні BigQuery та кешуванню. Продуктивність моніториться через інструменти Cloud Composer та Google Cloud Monitoring.

## 3.4 Вимоги бази даних.

Інтерфейс користувача аналітичної платформи буде реалізовано у вигляді вебзастосунку з інтерактивними дашбордами, створеними в Apache Superset. Основною метою є забезпечення швидкого та зручного доступу до аналітичної інформації для ключових ролей — маркетологів, продуктових аналітиків і менеджерів з продажу. Всі візуальні компоненти платформи будуть адаптовані до потреб прийняття рішень, спираючись на агреговані метрики та деталізовані зрізи даних.

Побудова інтерфейсу базується на принципі ієрархічної глибини: користувач спочатку отримує загальне уявлення про стан справ за допомогою ключових показників ефективності (KPI), а далі може перейти до вивчення часових трендів, категоріальних порівнянь і детальних таблиць. У верхній частині кожного дашборду буде розміщено короткий текстовий опис його призначення, що допомагає зорієнтуватися в аналітичному контексті. Графіки будуть згруповані за змістом у логічні блоки, кожен із яких позначено заголовком або маркуванням у markdown-блоці, що сприятиме структурованому сприйняттю інформації.

Особливу увагу буде приділено системі фільтрації: користувачі зможуть динамічно перемикатися між країнами, часовими періодами, категоріями товарів, брендами, SKU та типами пристроїв. Фільтри будуть реалізовані на рівні UI Superset без потреби писати SQL-запити. Завдяки використанню gold-моделей із попередньо агрегованими даними взаємодія з фільтрами буде швидкою й зручною навіть при великих обсягах інформації.

Ключовим елементом забезпечення актуальності інформації на дашбордах буде використання Apache Airflow. Цей інструмент дозволятиме автоматизувати оновлення даних: усі моделі будуть запускатися відповідно до розкладу через DAG-и, які контролюють порядок виконання задач, залежності між шаром моделей (bronze → silver → gold) та сповіщення про помилки. Це забезпечить синхронну оновлюваність усіх дашбордів і запобігатиме відображенню застарілої або частково оновленої інформації.

Інтерфейс дашбордів також буде витриманий у єдиному стилі з дотриманням принципів консистентного UI-дизайну: єдина палітра кольорів, чітка ієрархія шрифтів, узгоджена структура розташування елементів. Це зробить взаємодію з платформою інтуїтивною та приємною навіть для користувачів без глибокої технічної підготовки.

Таким чином, інтерфейс користувача виконуватиме не лише роль засобу візуалізації даних, а й стане повноцінним інструментом бізнес-аналітики, що дозволяє швидко реагувати на зміни в поведінці користувачів, ефективності маркетингових кампаній і результативності продажів.

## 3.5 Інші вимоги.

Вимоги до продуктивності (Performance Requirements):

* швидкість інжестування даних: Система повинна забезпечувати ефективне завантаження даних з джерел до шару Bronze BigQuery з мінімальною затримкою, витримуючи заплановані обсяги щоденних/щогодинних надходжень;
* швидкість трансформації даних: Час виконання трансформацій за допомогою dbt у BigQuery для оновлення шарів Silver та Gold має відповідати встановленим SLA для актуальності даних;
* час відповіді запитів BI: Запити до BigQuery з Apache Superset для відображення дашбордів та візуалізацій повинні мати час відповіді, що забезпечує комфортну роботу кінцевих користувачів;
* паралелізм завдань: Apache Airflow (Cloud Composer) повинен підтримувати виконання декількох завдань та DAG-ів паралельно, згідно з визначеною кількістю воркерів та їхніми ресурсами.

Вимоги до безпеки (Security Requirements):

* контроль доступу: Всі компоненти системи повинні інтегруватися з Google Cloud IAM для реалізації політик мінімальних привілеїв та контролю доступу на рівні користувачів та сервісних акаунтів;
* шифрування даних: Дані повинні бути зашифровані як під час передачі (encryption in transit - TLS/SSL), так і під час зберігання (encryption at rest) у BigQuery, Cloud Storage та Cloud SQL;
* аудит та логування: Система повинна генерувати детальні аудиторські логи всіх дій користувачів та системних подій, доступних для перегляду та аналізу через Google Cloud Logging;
* захист від несанкціонованого доступу: Зовнішні інтерфейси (наприклад, Superset UI, Airflow UI) повинні бути захищені за допомогою відповідних механізмів автентифікації та авторизації.

Вимоги до зручності використання (Usability Requirements):

* інтуїтивний інтерфейс BI: Веб-інтерфейс Apache Superset повинен бути інтуїтивно зрозумілим для кінцевих бізнес-користувачів, дозволяючи їм легко створювати та переглядати дашборди;
* зручність управління оркестрацією: Веб-інтерфейс Apache Airflow повинен надавати зручні інструменти для моніторингу DAG-ів, перегляду логів та управління завданнями для інженерів даних;
* документація для користувачів: Повинна бути надана чітка та зрозуміла документація для користувачів BI-інструменту та інженерів, що працюють з платформою.

Вимоги до сумісності (Compatibility Requirements):

* формати даних: Платформа повинна підтримувати роботу з різними форматами даних (наприклад, CSV, JSON, Parquet, Avro) як на етапі інжестування, так і для експорту;
* стандартні протоколи: Взаємодія з зовнішніми системами повинна базуватися на стандартних протоколах (наприклад, HTTPS, SQL);

Вимоги до документування (Documentation Requirements):

* технічна документація: Повна технічна документація архітектури, конфігурації, розгортання та керування платформою;
* документація коду: Код DAG-ів, моделей dbt та Terraform-ресурсів повинен бути добре прокоментований та відповідати стандартам кодування;
* документація метаданих: Необхідна документація щодо схем даних у BigQuery, визначення метрик та бізнес-логіки трансформацій.

Вимоги до вартості (Cost Requirements):

* платформа повинна бути розроблена з урахуванням оптимізації витрат на хмарні ресурси, використовуючи керовані сервіси та автоматичне масштабування для мінімізації надлишкових витрат.

# **4 ДОДАТКОВІ МАТЕРІАЛИ**

До специфікації додаються додаткові джерела (див.рис. 4.1– 4.4)

Зображення, що містить текст, схема, малюнок, коло

Вміст на основі ШІ може бути неправильним.

Рисунок 4.1 – Use-case діаграма

Зображення, що містить текст, схема, Шрифт, білий

Вміст на основі ШІ може бути неправильним.

Рисунок 4.2 – Діаграма синхронізації дагів

Зображення, що містить текст, знімок екрана, схема, ряд

Вміст на основі ШІ може бути неправильним.

Рисунок 4.3 – Data Flow Diagram

Рисунок 4.4 – Схема побудови датамартів Зображення, що містить текст, схема, знімок екрана, ряд

Вміст на основі ШІ може бути неправильним.