**Software Requirements**

**Specification**

**for**

**ArtGuard**

**Version 1.0 approved**

**Prepared by Viktor Novomlynskyi**

**NURE**

**06.05.2025**

**Історія версій**

| **Дата** | **Опис** | **Автор** | **Коментарі** |
| --- | --- | --- | --- |
| 06.05.2025 | Версія 1.0 | Новомлинський Віктор | Створення документу |

**Зміст**

[1. Вступ **4**](#_heading=h.qfpc4810kax7)

[1.1 Призначення документа 4](#_heading=h.qqaacl3eofd4)

[1.2 Область застосування 4](#_heading=h.tkudii9afq41)

[1.3 Цільова аудиторія 5](#_heading=h.o61szmvewsu2)

[1.4 Терміни, скорочення та абревіатури 6](#_heading=h.rroy202pzp5o)

[1.5 Огляд документа 7](#_heading=h.aurz2wv5yqt9)

[2. Загальний опис 7](#_heading=h.go9c5sa2tw1i)

[2.1 Перспектива продукту 7](#_heading=h.vbu4ve6010e9)

[2.2 Функціональність продукту 9](#_heading=h.xo0vm44jflg1)

[2.3 Характеристики користувачів 9](#_heading=h.bpfe6dgbwvzj)

[2.5 Загальні обмеження 12](#_heading=h.plymsnfgzzaq)

[3. Конкретні вимоги 13](#_heading=h.b05kbxi6bdxo)

[3.1 Вимоги до зовнішніх інтерфейсів 13](#_heading=h.ezrm36vbjgh8)

[3.1.1 Інтерфейси користувача 13](#_heading=h.y7ghxmqszet5)

[3.1.2 Апаратні інтерфейси 14](#_heading=h.4u61tivdb7le)

[3.1.3 Програмні інтерфейси 15](#_heading=h.pzztkofl0fax)

[3.1.4 Комунікаційні інтерфейси 16](#_heading=h.g3ru696lpbvm)

[3.2 Властивості програмного продукту 17](#_heading=h.2cekjemchqqa)

[3.3 Атрибути програмного продукту 18](#_heading=h.5rtujao9p4xf)

[3.3.1 Надійність 18](#_heading=h.gmkriyj334al)

[3.3.2 Доступність 19](#_heading=h.xpk70wc0m3mr)

[3.3.3 Безпека 19](#_heading=h.kdg23mtecnf4)

[3.3.4 Супровід 19](#_heading=h.8qoor3e92jrn)

[3.3.5 Переносимість 20](#_heading=h.68ixo4k7jrn2)

[3.3.6 Продуктивність 21](#_heading=h.mzercdq36phb)

[3.4 Вимоги до бази даних 21](#_heading=h.hewwkpoubmpb)

[3.5 Інші вимоги 22](#_heading=h.rj2zfn6n4daz)

[4. Додаткові матеріали 23](#_heading=h.dsmiug3flhp9)

[4.1 Діаграми баз даних 23](#_heading=h.sfjg987eh1)

[4.2 Діаграми прецедентів 25](#_heading=h.mz9cc28ysg3n)

[4.3 BPMN-діаграма 27](#_heading=h.mz9cc28ysg3n)

# 1. Вступ

Цей документ Software Requirements Specification (SRS) надає повний опис функціональних та нефункціональних вимог для розробки програмного продукту ArtGuard. Документ містить усю необхідну інформацію для інженерів-розробників, щоб спроєктувати та реалізувати програмне забезпечення відповідно до перелічених вимог. SRS створено відповідно до стандарту IEEE зі специфікацій програмних вимог і є основою для всіх етапів розробки, тестування та впровадження системи.

## 1.1 Призначення документа

Мета цього документа – надати докладний опис вимог до програмного забезпечення ArtGuard, призначеного для моніторингу умов зберігання музейних експонатів за допомогою IoT-технологій. Основна аудиторія цього документа включає інженерів-розробників, тестувальників, проектних менеджерів, технічну підтримку та потенційних користувачів з боку музеїв чи архівних установ. Цей SRS допоможе кожному учаснику розробки правильно зрозуміти вимоги та завдання, необхідні для успішної реалізації системи.

## 1.2 Область застосування

Продукт: ArtGuard — це система моніторингу умов зберігання музейних експонатів, яка збирає та аналізує дані з IoT-пристроїв (температура, вологість, вібрація), надсилає сповіщення при виявленні відхилень та забезпечує доступ до даних через мобільний і веб-застосунок. Основна ідея полягає в тому, щоб автоматизувати контроль мікроклімату для збереження об’єктів культурної спадщини, зменшуючи потребу в ручному моніторингу та людському факторі.

Функції:

* підключення IoT-сенсорів до серверної частини через Wi-Fi;
* прийом, обробка та зберігання даних (температура, вологість, вібрація) у реальному часі;
* веб-інтерфейс для перегляду даних, експонатів, локацій;
* мобільний застосунок для моніторингу стану об'єктів на ходу;
* система сповіщень про критичні відхилення параметрів;
* ролі користувачів: адміністратор, співробітник, переглядач;
* автентифікація з використанням Firebase або JWT;
* базова історія вимірювань (лог даних) з прив’язкою до об’єкта.

Застосування: ArtGuard буде корисним для музейних працівників, кураторів, технічного персоналу закладів культури, архівів або галерей, які потребують постійного контролю умов зберігання. Система дає змогу зменшити ризики пошкодження експонатів через несприятливі умови, своєчасно реагувати на відхилення та організовано працювати з даними з однієї централізованої платформи.

Цілі продукту:

* автоматизація моніторингу параметрів середовища;
* мінімізація впливу людського чинника на збереження експонатів;
* підвищення оперативності реагування у разі небезпеки;
* централізація даних з можливістю подальшої аналітики;
* зручний доступ до показників з будь-якого пристрою;
* гнучке масштабування (можливість додавання нових зон, сенсорів, користувачів).

## 1.3 Цільова аудиторія

Система ArtGuard орієнтована на установи, які потребують високоточний моніторинг кліматичних умов:

* державні та приватні музеї;
* художні галереї;
* виставкові центри;
* архіви з паперовими або текстильними матеріалами;
* бібліотеки з цінними фондами;
* приватні колекціонери;
* реставраційні центри.

Ці категорії користувачів потребують надійного, гнучкого, простого у використанні інструменту, що дозволяє зберігати чутливі об'єкти в оптимальних умовах.

## 1.4 Терміни, скорочення та абревіатури

SRS (Software Requirements Specification) – специфікація програмних вимог.

API (Application Programming Interface) – інтерфейс прикладного програмування.

ArtGuard – назва програмного продукту, описаного в цьому документі.

ESP32 – мікроконтролер із вбудованим Wi-Fi, що використовується як IoT-пристрій.

DHT22 – цифровий датчик температури та вологості.

MPU6050 – датчик вібрації/акселерометр.

JWT (JSON Web Token) – токен авторизації для захищеної взаємодії з API.

Firebase – платформа Google для автентифікації, авторизації та хмарного зберігання.

CRUD – набір базових операцій над даними: Create, Read, Update, Delete.

UI (User Interface) – інтерфейс користувача.

UX (User Experience) – досвід користувача.

DFD (Data Flow Diagram) – діаграма потоків даних.

STD (State Transition Diagram) – діаграма переходів станів.

DB – база даних.

IoT (Internet of Things) – інтернет речей: підключені пристрої, які збирають і передають дані.

## 1.5 Огляд документа

У наступному розділі цього документа, «Загальний опис», наведено огляд функціональних можливостей програмного продукту ArtGuard. Він описує неформальні вимоги до системи та використовується для встановлення контексту технічної специфікації, яка подається в наступному розділі.

Третій розділ цього документа, «Специфікація вимог», призначений насамперед для розробників і описує у технічних термінах функціональні та нефункціональні особливості системи.

Обидва розділи описують один і той самий програмний продукт, але адаптовані для різних типів читачів, тому використовують відповідно різний рівень деталізації.

У четвертому розділі подано моделі аналізу (діаграми), які допомагають структурувати вимоги та демонструють логіку взаємодії між компонентами системи.

# 2. Загальний опис

## 2.1 Перспектива продукту

ArtGuard — це система моніторингу, яка інтегрує IoT-пристрої, серверну інфраструктуру та клієнтські інтерфейси для контролю мікрокліматичних умов (температури, вологості, вібрацій) у зонах зберігання музейних експонатів.

Управління користувачами та автентифікація:

* F1: Реєстрація та вхід через Firebase або JWT. Користувачі можуть швидко створювати акаунти та входити до системи з будь-якого пристрою.
* F2: Ролі користувачів. Система підтримує розмежування прав доступу: адміністратор, співробітник, переглядач.

Моніторинг стану середовища:

* F3: Прийом показників від IoT-пристроїв (ESP32 + сенсори DHT22, MPU6050) у реальному часі.
* F4: Зберігання виміряних значень у базі даних PostgreSQL.
* F5: Перегляд історії замірів через веб- або мобільний інтерфейс.

Клієнтські інтерфейси:

* F6: Веб-інтерфейс (React) для перегляду об’єктів, сенсорів, сповіщень, керування таблицями (CRUD).
* F7: Мобільний застосунок (Kotlin + Jetpack Compose) для перегляду стану експонатів, деталей зон і параметрів.

Система сповіщень:

* F8: Генерація критичних сповіщень у випадку перевищення допустимих меж.
* F9: Можливість позначення сповіщень як "перевірені", додавання коментаря співробітником.

Локалізація та масштабування:

* F10: Підтримка мультимови через i18next (ua / en).
* F11: Розширення кількості сенсорів та об’єктів без переробки архітектури.

Аналітика та логування (майбутнє розширення):

* F12: Побудова графіків зміни параметрів експонатів за період.
* F13: Автоматичне формування звітів по зонах або об’єктах.
* F14: Інтеграція з системами зовнішнього моніторингу (Prometheus, Grafana).

## 2.2 Функціональність продукту

ArtGuard — це програмно-апаратна система, призначена для моніторингу мікрокліматичних параметрів у зонах зберігання музейних експонатів. Користувачі можуть переглядати поточні значення температури, вологості та вібрації, отримані від підключених IoT-сенсорів, а також аналізувати їх динаміку.

Система дозволяє налаштовувати критичні пороги для кожного типу сенсора, автоматично формує сповіщення про перевищення значень і зберігає дані для подальшого перегляду або аудиту. Доступ до інформації надається через веб-інтерфейс або мобільний застосунок.

ArtGuard також підтримує багаторівневу систему доступу з розподіленням ролей (адміністратор, співробітник, переглядач), що забезпечує безпечне керування об’єктами та доступом до даних. Сервіс адаптований до подальшого масштабування — передбачено можливість підключення додаткових сенсорів, розширення зони покриття, та інтеграцію з аналітичними системами.

Таким чином, ArtGuard не лише реєструє параметри середовища, а й допомагає установам культури забезпечувати контроль зберігання цінних об’єктів, попереджувати пошкодження та організовано управляти пов’язаною інформацією.

## 2.3 Характеристики користувачів

Користувачами системи ArtGuard є співробітники музейних, архівних або виставкових установ, відповідальні за збереження експонатів, а також системні адміністратори, які забезпечують технічну підтримку платформи.

Для ефективної роботи з системою користувачам бажано мати базові навички користування цифровими технологіями, а також розуміння принципів моніторингу мікрокліматичних умов.

Категорії користувачів.

Адміністратор:

* володіє повним доступом до системи;
* налаштовує об'єкти, зони, сенсори, користувачів;
* переглядає всі логи, сповіщення, конфігурації;
* має базові знання у сфері ІТ та адміністрування.

Співробітник музею (Staff):

* має доступ до перегляду експонатів і сенсорів у закріплених зонах;
* може позначати сповіщення як опрацьовані;
* не змінює конфігурацію системи;
* здатен користуватись мобільним/веб застосунком на базовому рівні.

Переглядач (Viewer):

* має лише режим читання: перегляд значень, історії, об’єктів;
* не може вносити зміни чи залишати коментарі.

**2.4 Загальні обмеження**

У межах реалізації курсового проєкту ArtGuard необхідно врахувати такі загальні обмеження:

1. Платформи та інтерфейси::
   * система реалізується у вигляді веб-додатку (на React) та мобільного застосунку (на Kotlin з Jetpack Compose);
   * клієнтські інтерфейси адаптовані для перегляду інформації на десктопі та мобільних пристроях;
   * обов’язкове забезпечення стабільної роботи в сучасних браузерах (Chrome, Firefox, Safari) та на Android-пристроях;
   * у рамках курсового проєкту реалізовано лише базову тематику інтерфейсу, без перемикання між світлим і темним режимами.
2. Функціональні обмеження:
   * реалізовано базову автентифікацію користувачів із підтримкою ролей (адміністратор, співробітник, переглядач), однак без глибокої системи прав доступу;
   * історія замірів зберігається, але не включає повного журналу подій або аудиту змін;
   * сповіщення формуються всередині інтерфейсу, без зовнішніх каналів (SMS, e-mail, push);
   * графічна аналітика обмежується статичним переглядом показників та діаграм.
3. Технічні обмеження:
   * передача даних з IoT-пристроїв (ESP32) здійснюється через HTTP-запити;
   * автоматичне виявлення нових пристроїв не реалізоване — конфігурація здійснюється вручну;
   * резервне копіювання бази даних передбачається як зовнішня опція, не інтегрована в систему;
   * система не підтримує офлайн-режим роботи клієнтів.

Ці обмеження пояснюються масштабом курсового проєкту, термінами його виконання та цільовою демонстрацією основної ідеї системи моніторингу в реальному або симульованому середовищі.

## 2.5 Загальні обмеження

Припущення:

* + користувачами системи є співробітники музеїв, архівів або інших закладів, які мають базові навички користування цифровими інструментами та розуміють важливість контролю умов зберігання експонатів;
  + передбачається, що система буде використовуватись у приміщеннях зі стабільним доступом до інтернету (Wi-Fi);
  + користувачі працюють з сучасними браузерами (Chrome, Firefox, Safari) або Android-пристроями з актуальною версією ОС;
  + всі підключення IoT-пристроїв (ESP32) здійснюються вручну або за допомогою заздалегідь визначеної інструкції;
  + обслуговування серверної частини здійснюється локально або в рамках внутрішньої мережі закладу, без потреби у хмарній інфраструктурі.

Залежності:

* + система залежить від стабільного функціонування IoT-сенсорів (DHT22, MPU6050) для збору даних у режимі близькому до реального часу;
  + вся інформація зберігається у реляційній базі даних PostgreSQL, до якої мають доступ клієнтські додатки через REST API;
  + авторизація користувачів реалізована через JWT або Firebase Authentication, що потребує відповідної інтеграції на стороні серверу;
  + надходження даних від сенсорів базується на регулярних HTTP-запитах від ESP32 — у разі відсутності сигналу або Wi-Fi зв’язку моніторинг припиняється;
  + подальший розвиток системи може вимагати підключення зовнішніх сервісів аналітики (наприклад, Grafana або Prometheus) — але наразі ця інтеграція не передбачена в межах курсового проєкту.

# 3. Конкретні вимоги

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейси користувача

Інтерфейс користувача в системі ArtGuard реалізовано у вигляді односторінкового веб-застосунку (SPA) на базі React, а також окремого мобільного застосунку, створеного на Kotlin з використанням Jetpack Compose. Обидва інтерфейси мають сучасний, мінімалістичний дизайн, адаптований під десктопні та мобільні пристрої.

Користувацькі інтерфейси включають:

* + головна панель (дашборд), відображає загальний стан об’єктів і зон; виводить список активних сенсорів, їх останні значення, кнопки швидкого переходу до картки об'єкта, історії показників та сповіщень.
  + сторінка входу, містить форму авторизації з полями логін/пароль, а також можливість входу через Firebase (Google-акаунт).
  + інтерфейс перегляду об’єкта, показує загальну інформацію про експонат: назву, опис, місцезнаходження, список прикріплених сенсорів, поточні та останні значення параметрів середовища.
  + історія показників, відображає табличну або графічну форму записаних значень з датами, можливістю фільтрації за періодом часу чи типом сенсора.
  + сторінка сповіщень, виводить список усіх активних або закритих сповіщень з позначками критичності, датами, статусом та можливістю залишити коментар.
  + налаштування профілю, користувач може змінювати свої дані, обрати мову інтерфейсу, активувати/деактивувати сповіщення, видалити обліковий запис.
  + адаптивність, веб-застосунок реалізовано з використанням CSS Grid / Flexbox, забезпечено підтримку сучасних екранів та мобільної верстки. Темна тема планується як опціональне розширення.

### 3.1.2 Апаратні інтерфейси

Система ArtGuard підтримується в усіх сучасних браузерах на персональних комп’ютерах і мобільних пристроях.

* + десктопи з операційними системами Windows, macOS або Linux;
  + смартфони та планшети на Android (версія 9 і вище);
  + мінімальна рекомендована ширина екрана: 360 пікселів.

Серверна частина:

Серверна частина системи розгортається на будь-якому віртуальному або фізичному сервері, який підтримує виконання застосунків на Go (Go runtime environment). Для зберігання даних використовується СУБД PostgreSQL.

* + підтримка REST API для обробки запитів від клієнтів і сенсорів;
  + обсяг оброблюваних даних розрахований на середнє навантаження (до кількох десятків сенсорів одночасно);
  + підтримка взаємодії з ESP32-пристроями через HTTP-запити.

IoT-пристрої:

* + мікроконтролери ESP32 з підключенням до Wi-Fi;
  + сенсори DHT22 (температура та вологість) і MPU6050 (вібрації);
  + обмін даними: періодичні HTTP POST-запити до API сервера.

### 3.1.3 Програмні інтерфейси

Авторизація користувачів:

* + реалізована через Firebase Authentication або за допомогою JWT (JSON Web Token);
  + протокол обміну: HTTPS;
  + вихідні дані: унікальний ID користувача, email, роль, токен доступу;
  + використовуються бібліотеки Firebase SDK (для мобільного застосунку) або middleware для Go (для веб).

Бекенд API:

* + серверна частина реалізована на мові Go з використанням фреймворку Fiber;
  + REST API підтримує запити типу GET, POST, PUT, DELETE;
  + передача даних відбувається у форматі JSON;
  + автентифікація запитів — через JWT (Authorization Header);
  + основні ендпоінти: /auth, /sensors, /readings, /alerts, /objects, /users.

База даних:

* + система використовує PostgreSQL для зберігання даних;
  + взаємодія із БД реалізована через бібліотеку GORM або аналогічний ORM для Go;

Основні таблиці:

* users — таблиця для зберігання облікових записів користувачів із зазначенням ролі (admin, user), а також даних для автентифікації;
* museums — містить інформацію про музейні установи, кожна з яких може мати кілька зон зберігання;
* zones — визначає логічні зони всередині музеїв із окремими кліматичними параметрами. Зона належить одному музею;
* objects — описує мистецькі об’єкти, що зберігаються у певній зоні. Один арт-об’єкт належить до однієї зони;
* sensors — зберігає дані про сенсори (тип, розташування, активність), кожен з яких прив’язаний до певної зони;
* measurements — таблиця з історією вимірювань від сенсорів, включаючи показники температури, вологості, вібрації та час запису;
* thresholds — встановлює межі допустимих значень кліматичних параметрів для кожної зони. Зв’язується із сенсорами;
* alerts — фіксує автоматично згенеровані сповіщення про перевищення порогових значень. Зв’язується з вимірюваннями та користувачем, який їх отримав.

Збір даних з сенсорів:

* + ESP32 надсилає дані через HTTP-запити безпосередньо на сервер (/api/measurements);
  + підтримується ручне або автоматичне підключення сенсорів із заданим ідентифікатором;
  + дані надсилаються у вигляді JSON-об’єктів, що містять тип сенсора, значення, час, ID зони або об’єкта.

### 3.1.4 Комунікаційні інтерфейси

Типи з’єднань:

* + усі запити від веб- та мобільного клієнта до серверної частини системи ArtGuard надсилаються через протокол HTTP;
  + клієнтські додатки (React, Kotlin) обмінюються даними із сервером через REST API;
  + IoT-пристрої (ESP32) також надсилають HTTP POST-запити на API для передачі показників з сенсорів.

Формат даних:

* + уся взаємодія між клієнтами, сервером та сенсорами реалізована з використанням формату JSON;
  + стандартна структура відповіді включає статус запиту, повідомлення, а також об'єкт або список об’єктів.

Безпека:

* + комунікація здійснюється через TLS версії 1.2 або вище;
  + для автентифікації користувачів застосовується передача JWT-токенів у заголовку запиту (Authorization: Bearer <token>);
  + серверна частина проводить валідацію вхідних даних (тип, діапазон, структура) з обов’язковим логуванням помилок.

У перспективі:

* + можлива реалізація WebSocket-з’єднань для передачі оновлень даних у реальному часі (наприклад, для інтерактивного моніторингу);
  + інтеграція з хмарними сервісами для централізованої аналітики або масштабування повідомлень (email / push через сторонні API).

## 3.2 Властивості програмного продукту

Програмний продукт ArtGuard розроблено як багаторівнева клієнт-серверна система, яка забезпечує централізоване збирання, зберігання, обробку та відображення даних про мікрокліматичні умови у середовищах зберігання музейних експонатів. Властивості програмного забезпечення базуються на потребах стабільного моніторингу, доступу до інформації у реальному часі та можливості масштабування у майбутньому.

Система складається з окремих незалежних компонентів:

* + серверна частина (Go), що відповідає за обробку логіки, автентифікацію, REST API та роботу з базою даних;
  + база даних PostgreSQL — для надійного та структурованого зберігання інформації про об’єкти, зони, сенсори, показники;
  + фронтенд (React) — для адміністрування, перегляду даних, керування обліковими записами;
  + мобільний додаток (Kotlin) — для швидкого доступу до ключової інформації зі смартфона;
  + IoT-рівень (ESP32 + сенсори DHT22, MPU6050) — для збору фізичних параметрів середовища.

## 3.3 Атрибути програмного продукту

Програмний продукт ArtGuard має відповідати сучасним вимогам до надійності, доступності, безпеки, сумісності та продуктивності систем, що функціонують у реальному середовищі із залученням фізичних пристроїв (IoT), баз даних та користувацьких інтерфейсів.

### 3.3.1 Надійність

Надійність системи полягає у здатності ArtGuard безперервно функціонувати протягом тривалого часу, виявляти і правильно обробляти нестандартні ситуації (відсутність даних, обрив підключення сенсора, спроби некоректного доступу тощо). Передбачено механізми автоматичного виявлення втрати зв’язку з сенсором, генерацію службових повідомлень, а також журналювання критичних подій. Рівень надійності забезпечується за рахунок стабільної логіки обробки даних, системи дублювання критичних дій (підтвердження, перевірка типу тощо), та незалежної передачі даних з сенсорів.

### 3.3.2 Доступність

Система повинна бути доступною для користувачів у режимі 24/7 через веб-інтерфейс або мобільний застосунок. Особлива увага приділяється наданню оперативного доступу до ключової інформації (стан зон, нові сповіщення, поточні значення сенсорів). Мобільна версія дозволяє персоналу музею працювати без прив’язки до конкретного місця. Для відображення критичної інформації не потрібне глибоке перемикання між розділами.

### 3.3.3 Безпека

Усі дані передаються через зашифровані канали зв’язку (HTTPS). Доступ до функціоналу обмежений через систему ролей (адміністратор, співробітник, переглядач) з автентифікацією через JWT або Firebase. Захист від несанкціонованого доступу реалізовано на рівні backend (middleware перевірка токена), з обов’язковим логуванням ключових подій. Користувачі можуть працювати тільки у межах дозволених зон і не мають доступу до налаштувань за межами своєї компетенції.

### 3.3.4 Супровід

Система ArtGuard розроблена з урахуванням принципів супровідності, що забезпечує легке внесення змін до програмного коду, конфігурацій та структури бази даних. Архітектура проєкту побудована модульно: окремі частини відповідають за авторизацію, роботу з сенсорами, об’єктами, користувачами та сповіщеннями. Це дозволяє змінювати чи вдосконалювати окремі компоненти без впливу на інші частини системи. Серверна логіка реалізована мовою Go, із застосуванням фреймворку Fiber, що забезпечує зрозумілу структуру маршрутів та middleware. Для роботи з базою даних використовується GORM — ORM, який дозволяє швидко оновлювати схеми та підтримує міграції.

Коментарі в коді, зрозумілі назви змінних, розділення логіки за папками — усе це спрощує супровід системи, її адаптацію під інші заклади або нові функції, включно з аналітичними модулями, сповіщеннями чи розширеним аудитом.

### 3.3.5 Переносимість

ArtGuard є переносимим рішенням, яке може бути встановлене та запущене на різних програмних і апаратних платформах без значної модифікації. Завдяки використанню:

* + вебтехнологій (React, REST API, PostgreSQL);
  + універсального бекенду (Go);
  + стандартного обміну даними у форматі JSON;

Cистему можна легко розгорнути на різних серверних середовищах — локально, у хмарі (VPS), або навіть у контейнерах (Docker).

Також мобільний застосунок створено під Android, але структура його коду дозволяє при потребі перенести логіку на iOS через мультиплатформенні рішення (наприклад, Kotlin Multiplatform чи Flutter, якщо буде реалізовано у майбутньому).

Передбачено, що у разі зміни сенсорного обладнання (інший тип датчика), програмна частина зможе адаптуватися з мінімальними змінами завдяки відкритій структурі API для IoT.

### 3.3.6 Продуктивність

Система повинна обробляти дані від кількох десятків сенсорів одночасно із затримкою не більше 2–3 секунд на оновлення показників. Серверна частина оптимізована під легковагові запити з сенсорів та клієнтських додатків. Передбачається можливість масштабування до 100+ сенсорів без суттєвих змін архітектури. Операції в базі даних (вибірка, фільтрація, запис показників) не перевищують стандартного порогу 100–150 мс у тестовому середовищі.

## 3.4 Вимоги до бази даних

Система ArtGuard використовує реляційну базу даних PostgreSQL для зберігання всієї основної інформації, що необхідна для повноцінного функціонування програмного забезпечення. База даних повинна забезпечувати цілісність, надійність та швидкий доступ до таких типів даних:

* + користувачі: облікові записи з ролями, автентифікаційні дані (email, токен доступу), історія дій;
  + музеї та зони: інформація про логічні зони зберігання та їх зв'язок з музейними об'єктами;
  + об’єкти: назва, опис, тип, прив’язка до зони;
  + сенсори: тип, унікальний ідентифікатор, прив’язка до зони, статус активності;
  + вимірювання (measurements): температура, вологість, вібрація, час запису, сенсор, зона;
  + пороги (thresholds): допустимі межі кліматичних показників для кожної зони;
  + сповіщення (alerts): автоматично згенеровані записи про перевищення меж з посиланням на вимірювання та користувача.

До бази даних висуваються наступні вимоги:

* + підтримка транзакцій та цілісності даних (ACID);
  + швидка обробка запитів вибірки (менше 150 мс у межах локального середовища);
  + регулярне резервне копіювання (відповідальність адміністратора);
  + можливість масштабування обсягу зберіганих даних із зростанням кількості сенсорів або об’єктів.

## 3.5 Інші вимоги

Часова зона: усі записи часу повинні зберігатися у форматі UTC з локальним перетворенням у інтерфейсі.

Журналювання подій: критичні дії (зміна порогів, додавання сенсора, створення об’єкта) мають зберігатися в логах з міткою часу, ID користувача та описом події.

Резервування і масштабування: система повинна мати можливість масштабування (більше зон, більше пристроїв, додаткові ролі) без потреби змінювати основну архітектуру.

Обмеження доступу: реалізовано контроль прав доступу до кожної функції залежно від ролі користувача.

Можливість інтеграції: система має бути здатною до подальшої інтеграції з платформами збору аналітики або виводу даних (наприклад, Grafana).

# 4. Додаткові матеріали

Цей розділ містить графічні матеріали, які допомагають краще зрозуміти структуру та логіку роботи програмного продукту ArtGuard. Діаграми бази даних, прецедентів та бізнес-процесів служать для візуалізації моделі даних, поведінки користувачів у системі та взаємодії функціональних компонентів.

## 4.1 Діаграми баз даних

У системі ArtGuard використовується реляційна база даних, побудована на основі таких логічних сутностей:

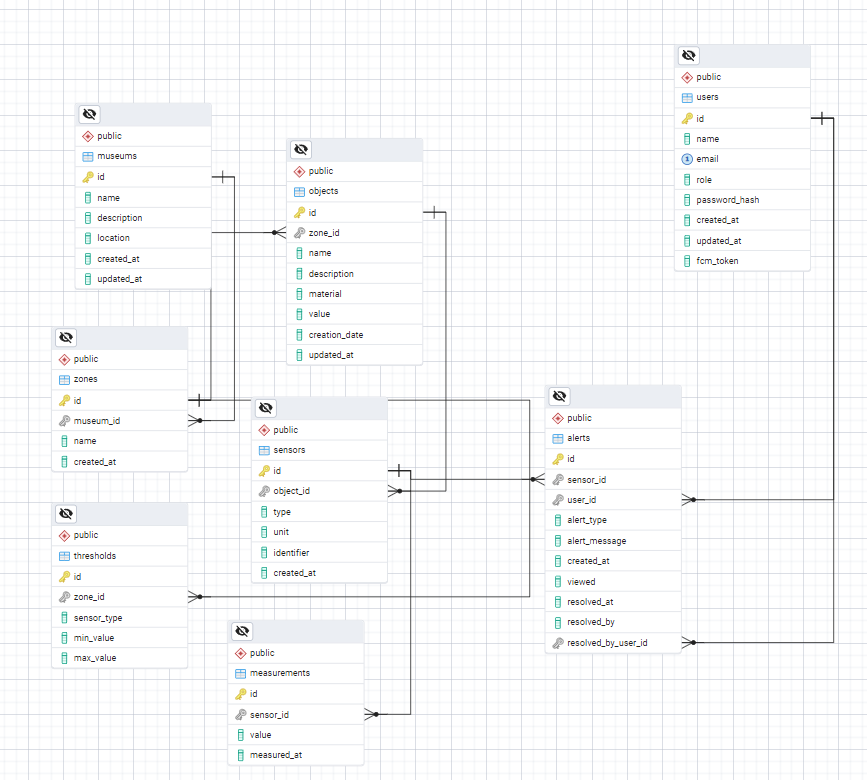
****

Рисунок 4.1 – Схема бази даних системи ArtGuard

* users – зберігає облікові записи користувачів, ролі, контактну інформацію;
* museums – інформація про музейні установи, що підключені до системи;
* zones – зони контролю всередині музею, до яких прикріплені сенсори;
* objects – мистецькі об’єкти, що фізично знаходяться в конкретній зоні;
* sensors – апаратні пристрої з унікальним ID, типом і належністю до зони;
* measurements – записи про всі зібрані показники: температура, вологість, вібрація;
* thresholds – допустимі межі параметрів для кожної зони;
* alerts – сповіщення, згенеровані системою при перевищенні порогових значень.

Схема бази даних відображає структуру зв’язків між основними сутностями системи ArtGuard. Кожна таблиця має чітке призначення та пов’язана з іншими через зовнішні ключі, що забезпечує цілісність даних і логічну ієрархію.

Один музей (museums) може містити кілька зон (zones), кожна з яких має власні параметри моніторингу.

У зоні можуть розміщуватись кілька об’єктів (objects) та сенсорів (sensors).

Сенсори генерують вимірювання (measurements), які фіксуються із зазначенням часу, типу значення та зони.

Якщо значення виходить за межі допустимих параметрів (thresholds), система створює сповіщення (alerts).

Облікові записи користувачів (users) зв’язуються зі сповіщеннями — наприклад, той, хто їх переглянув або обробив.

Така структура дозволяє масштабувати систему: додавати нові об’єкти, зони, сенсори, не змінюючи архітектуру. Усі дані організовані у логічні групи, що забезпечує високу зрозумілість та ефективну роботу системи в реальному середовищі.

## 4.2 Діаграми прецедентів

Діаграми прецедентів (Use Case diagrams) демонструють взаємодію користувачів системи ArtGuard з її функціональністю через інтерфейс. Кожен прецедент (дія) представляє собою окремий сценарій, доступний певному типу користувача, залежно від його ролі.

На рисунку 4.2 зображено діаграму прецедентів для веб-інтерфейсу ArtGuard, який використовується адміністраторами та переглядачами.

Основні сценарії:

* входу в систему;
* перегляду головної сторінки;
* перемикання мови (ua/en);
* перегляду каруселі картин;
* резервного копіювання.

Адміністратор додатково має доступ до:

* адмін-панелі;
* зміни формату дати/часу;
* CRUD-операцій з таблицями;
* валідації імпорту.

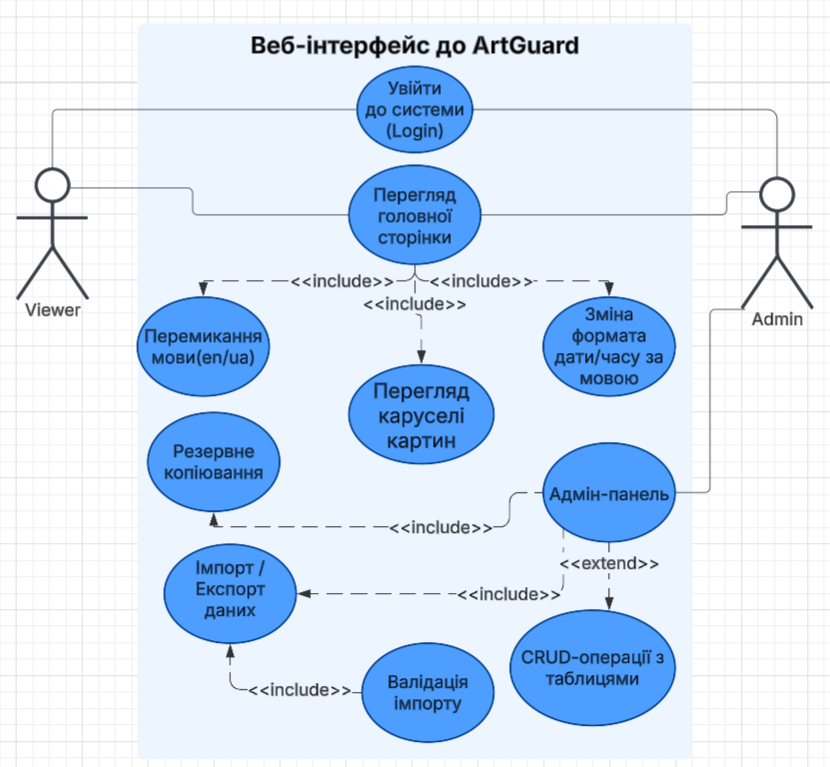
****

Рисунок 4.2 – Діаграма прецедентів для веб-інтерфейсу ArtGuard

Усі функції включено до перегляду головної сторінки за допомогою <<include>>, а доступ до CRUD-операцій розширюється з адмін-панелі за допомогою <<extend>>.

## 4.3 BPMN-діаграма

На рисунку 4.3 представлено BPMN-діаграму процесу моніторингу кліматичних умов у зоні зберігання музейних експонатів, реалізованого в системі ArtGuard. Діаграма включає всі ключові етапи обробки даних: від збору сенсорами до реагування персоналу та адміністратора.

Основні учасники процесу:

* Сенсор (ESP32) — виконує збір та передачу показників (температура, вологість, вібрація);
* Сервер ArtGuard — зберігає дані, перевіряє їх відповідність нормам, формує сповіщення;
* Працівник музею — отримує критичні сповіщення, переглядає деталі та фіксує реакцію;
* Адміністратор — аналізує журнал подій та, за потреби, змінює порогові значення.

Логіка процесу:

* Після збору даних сенсором вони передаються на сервер.
* Сервер зберігає показники та перевіряє їх на відповідність порогам.
* У разі відхилень формується сповіщення.
* Працівник отримує це сповіщення, переглядає деталі та позначає його як опрацьоване.
* Адміністратор перевіряє частоту подібних подій і коригує порогові межі.

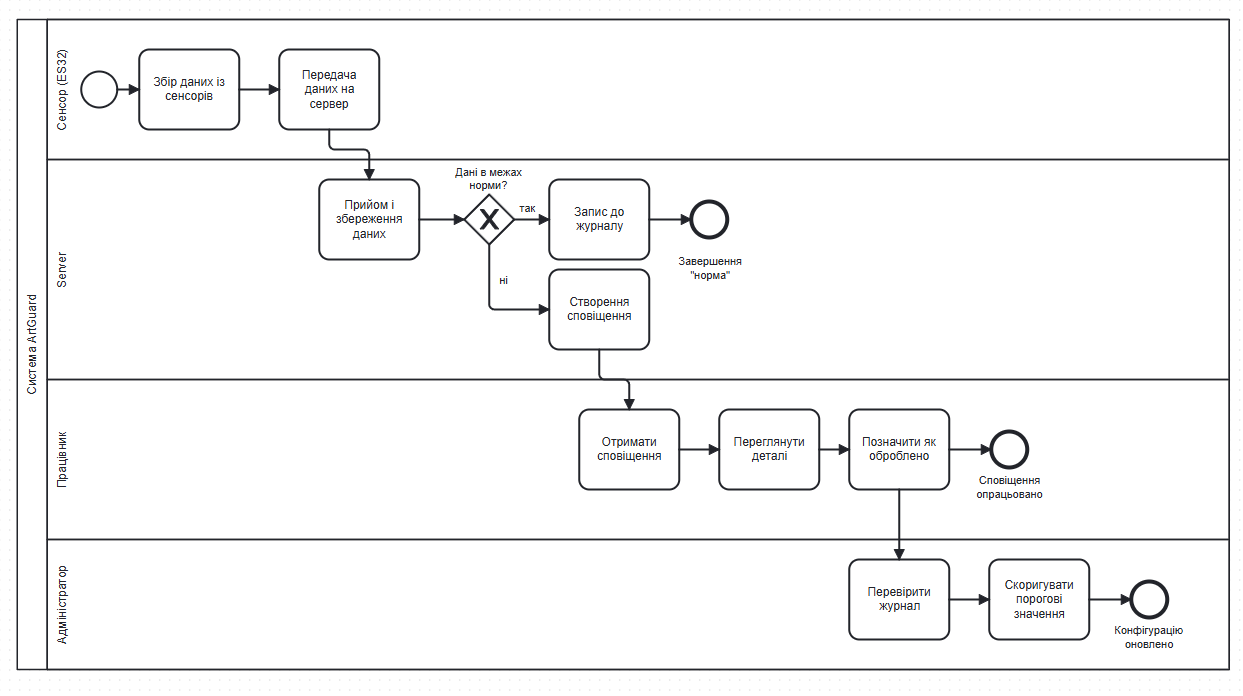
****

Рисунок 4.3 – Діаграма BPMN процесу моніторингу кліматичних умов у зоні зберігання