SoilScout

Software Requirements Specification

Version 1

06.06.2025

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 06.06.2025 | Version 1 | Щербатюк Максим | First Revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Щербатюк Максим |  | 06.06.2025 |
|  |  |  |  |

**Table of Contents**

[1. Introduction 5](#_Toc201107065)

[1.1 Purpose 5](#_Toc201107066)

[1.2 Scope 5](#_Toc201107067)

[1.3 Definitions, Acronyms, and Abbreviations 7](#_Toc201107068)

[1.4 References 9](#_Toc201107069)

[1.5 Overview 10](#_Toc201107070)

[2. General Description 12](#_Toc201107071)

[2.1 Product Perspective 12](#_Toc201107072)

[2.2 Product Functions 13](#_Toc201107073)

[2.3 User Characteristics 13](#_Toc201107074)

[2.4 General Constraints 13](#_Toc201107075)

[2.5 Assumptions and Dependencies 13](#_Toc201107076)

[3. Specific Requirements 15](#_Toc201107077)

[3.1 External Interface Requirements 15](#_Toc201107078)

[3.1.1 User Interfaces 15](#_Toc201107079)

[3.1.2 Hardware Interfaces 15](#_Toc201107080)

[3.1.3 Software Interfaces 15](#_Toc201107081)

[3.1.4 Communications Interfaces 16](#_Toc201107082)

[3.2 Functional Requirements 16](#_Toc201107083)

[3.2.1 Реєстрація користувача 16](#_Toc201107084)

[3.2.2 Авторизація користувача 17](#_Toc201107085)

[3.2.3 Перегляд сторінки профілю 17](#_Toc201107086)

[3.2.4 Редагування профілю 18](#_Toc201107087)

[3.2.5 Вихід з профілю 19](#_Toc201107088)

[3.2.6 Перегляд списку полів 19](#_Toc201107089)

[3.2.7 Пошук полів 20](#_Toc201107090)

[3.2.8 Сортування полів 20](#_Toc201107091)

[3.2.9 Фільтрація полів за активністю 21](#_Toc201107092)

[3.2.10 Створення поля 21](#_Toc201107093)

[3.2.11 Редагування поля 22](#_Toc201107094)

[3.2.12 Видалення поля 23](#_Toc201107095)

[3.2.13 Перегляд сторінки поля 23](#_Toc201107096)

[3.2.14 Передача даних з IoT-пристрою 24](#_Toc201107097)

[3.2.15 Перегляд деталей точки (мобільний додаток) 25](#_Toc201107098)

[3.2.16 Отримання сповіщень (мобільний додаток) 25](#_Toc201107099)

[3.2.17 Перегляд усіх сповіщень (мобільний додаток) 26](#_Toc201107100)

[3.2.18 Видалення всіх сповіщень (мобільний додаток) 26](#_Toc201107101)

[3.2.19 Перегляд аналітики поля (веб-платформа) 27](#_Toc201107102)

[3.2.20 Керування користувачами (для адміністраторів) 27](#_Toc201107103)

[3.2.21 Керування девайсами (для адміністраторів) 28](#_Toc201107104)

[3.2.22 Керування сенсорами (для адміністраторів) 29](#_Toc201107105)

[3.2.23 Керування пристроями користувачів (для адміністраторів) 29](#_Toc201107106)

[3.3 Non-Functional Requirements 30](#_Toc201107107)

[3.3.1 Performance 30](#_Toc201107108)

[3.3.2 Reliability 30](#_Toc201107109)

[3.3.4 Security 31](#_Toc201107110)

[3.3.6 Portability 31](#_Toc201107111)

[3.4 Classes / Objects 31](#_Toc201107112)

[3.4.1 field\_measurements 31](#_Toc201107113)

[3.4.2 fields 31](#_Toc201107114)

[3.4.3 iot\_devices 32](#_Toc201107115)

[3.4.4 measurement\_points 32](#_Toc201107116)

[3.4.5 measurements 33](#_Toc201107117)

[3.4.6 refresh\_sessions 33](#_Toc201107118)

[3.4.7 sensors 33](#_Toc201107119)

[3.4.8 user\_iot\_devices 33](#_Toc201107120)

[3.4.9 users 34](#_Toc201107121)

[3.6 Inverse Requirements 34](#_Toc201107122)

[3.7 Design Constraints 35](#_Toc201107123)

[3.8 Other Requirements 35](#_Toc201107124)

[4. Analysis Models 36](#_Toc201107125)

[4.1 Use Case Diagrams 36](#_Toc201107126)

[4.1.1 Use Case android частини системи 36](#_Toc201107127)

[4.1.2 Use Case web частини системи 37](#_Toc201107128)

[4.2 Database Design 37](#_Toc201107129)

[4.2.1 Logical Database Schema 37](#_Toc201107130)

[4.3 Sequence Diagrams 38](#_Toc201107131)

[4.4 State-Transition Diagrams (STD) 38](#_Toc201107132)

[4.5 Data Flow Diagrams (DFD) 38](#_Toc201107133)

[5. Change Management Process 38](#_Toc201107134)

# 1. Introduction

## 1.1 Purpose

Цей документ, є офіційним документом, що детально описує функціональні та нефункціональні вимоги до програмної системи моніторингу стану сільськогосподарських культур (SoilScout). Основною метою цього SRS є створення єдиного, чіткого та всебічного розуміння системи для всіх зацікавлених сторін. Це включає розробників, тестувальників, керівників проєкту та кінцевих замовників. Документ слугуватиме основою для подальших етапів розробки, включаючи проектування, реалізацію, тестування та впровадження системи.

SRS SoilScout розроблено для досягнення наступних цілей:

* **Чітке визначення потреб:** Забезпечити повний і точний опис того, що має робити система, відповідаючи на питання "що", а не "як".
* **Спільне розуміння:** Усунути неоднозначності та розбіжності у трактуванні вимог між замовником та командою розробки, гарантуючи, що всі сторони працюють над однією і тією ж метою.
* **Основа для розробки та тестування:** Слугувати вихідною точкою для архітектурного проектування, кодування та розробки планів тестування, дозволяючи тестувальникам ефективно перевіряти відповідність реалізованого функціоналу вимогам.
* **Інструмент управління проєктом:** Забезпечити основу для оцінки обсягу робіт, планування ресурсів та моніторингу прогресу проєкту. Будь-які зміни у вимогах, виявлені після затвердження цього SRS, будуть проходити через формальний процес управління змінами, щоб мінімізувати ризики та вплив на проєкт.
* **Зниження ризиків:** Виявлення потенційних проблем та суперечностей у вимогах на ранніх етапах розробки, що дозволяє уникнути дорогих переробок на пізніших стадіях.

Цей документ буде періодично переглядатися та оновлюватися відповідно до змін у вимогах або в бізнес-середовищі, згідно з визначеним процесом управління змінами.

## 1.2 Scope

Система SoilScout – це інноваційна, комплексна інформаційна система, розроблена для глибокого моніторингу та ефективного управління станом ґрунту на сільськогосподарських полях. Вона інтегрує передові технології IoT (Інтернету речей) з потужною веб-платформою та зручним мобільним додатком для Android, створюючи єдиний, цілісний інструмент для сучасного агронома та фермера. Основна мета системи – надати аграріям (від великих холдингів до приватних фермерів) точні, актуальні та легкодоступні дані про ключові параметри ґрунту, що дозволить підвищити ефективність сільськогосподарського виробництва, оптимізувати використання ресурсів та зменшити негативний вплив на довкілля.

**Система охоплює наступні ключові функціональні області:**

* **Моніторинг параметрів ґрунту:** Центральною функцією є збір даних про температуру, вологість та кислотність ґрунту. Ці дані збираються за допомогою спеціально розробленого IoT-пристрою, який забезпечує високу точність вимірювань.
* **Візуалізація та аналітика даних:** Зібрані дані обробляються та відображаються у зручному та зрозумілому форматі. Користувачі можуть переглядати показники у вигляді графіків, таблиць та інтерактивних карт на веб-платформі та в мобільному додатку, що дозволяє швидко оцінити стан ґрунту та виявити аномалії.
* **Управління полями та точками вимірювань:** Система надає широкі можливості для управління сільськогосподарськими полями. Користувачі можуть створювати, редагувати та видаляти інформацію про свої поля, а також прив'язувати до них дані, отримані з IoT-пристроїв. Це дозволяє організувати дані відповідно до реальних об'єктів на місцевості.
* **Система сповіщень:** Для своєчасного реагування на зміни, система передбачає механізм сповіщень. Користувачі отримуватимуть оповіщення про критичні показники ґрунту (наприклад, низька вологість або невідповідний рівень pH), що дозволить оперативно вживати необхідні заходи.
* **Безпека та управління доступом:** Реалізовано надійну систему автентифікації та авторизації, що забезпечує безпеку даних користувачів. Розмежування прав доступу на основі ролей (наприклад, фермер, агроном, адміністратор) гарантує, що кожен користувач має доступ лише до тієї інформації та функцій, які відповідають його ролі.
* **Багатомовність:** З метою охоплення ширшої аудиторії та зручності використання, як веб-інтерфейс, так і мобільний додаток підтримують дві мови: українську та англійську.

**Цілі початкового випуску (MVP):**

* Забезпечення надійного збору, зберігання та відображення базових показників ґрунту (температура, вологість, кислотність) з IoT-пристрою.
* Реалізація повного CRUD-функціоналу (створення, читання, оновлення, видалення) для інформації про поля та пов'язаних з ними вимірювань.
* Повноцінний функціонал реєстрації, входу та управління профілем користувача, включаючи підтримку автентифікації через електронну пошту/пароль та Google-акаунт.
* Візуалізація даних у вигляді інформативних графіків (часові ряди) та зручних таблиць для глибокого аналізу.
* Надійна та повноцінна двомовна підтримка (українська та англійська) для всіх елементів користувацького інтерфейсу на веб-платформі та в мобільному додатку.

**Обсяг наступних випусків та потенційні розширення системи:**

* **Інтеграція з метеорологічними даними:** Додавання можливості інтеграції з локальними метеорологічними станціями або зовнішніми API для отримання даних про опади, температуру повітря, вологість повітря та інші метеопараметри. Це дозволить здійснювати більш комплексний аналіз взаємозв'язків між погодними умовами та станом ґрунту.
* **Рекомендації щодо внесення добрив:** Розробка модуля для надання автоматизованих рекомендацій щодо типів та кількості добрив на основі аналізу зібраних даних про ґрунт та потреб конкретних сільськогосподарських культур.
* **Прогнозування стану ґрунту:** Впровадження алгоритмів машинного навчання для прогнозування змін у стані ґрунту, таких як ймовірність посухи або дефіциту поживних речовин.
* **Інтеграція з іншим сільськогосподарським обладнанням:** Можливість підключення до інших типів агротехнічного обладнання для автоматизованого збору додаткових даних або управління процесами.
* **Автоматизований збір даних за допомогою дронів:** Розширення функціоналу для інтеграції з даними, отриманими з агродронів (наприклад, мультиспектральні знімки), для створення більш точних карт полів та моніторингу стану посівів.
* **Кросплатформенність мобільного додатку:** Розширення мобільного додатку для підтримки операційної системи iOS, що дозволить охопити ширшу аудиторію користувачів.

**Обмеження системи:**

* Користувач має доступ до функцій системи лише якщо має IoT-девайс.
* Користувач має доступ до мобільного застосунку, якщо зареєструвався на веб-сайті.

## 

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| IoT (Internet of Things) | Інтернет речей – мережа фізичних об'єктів, оснащених сенсорами, програмним забезпеченням та іншими технологіями, що дозволяють їм підключатися та обмінюватися даними з іншими пристроями та системами через Інтернет. У контексті SoilScout, це пристрій для вимірювання параметрів ґрунту. |
| RESTful API | Інтерфейс прикладного програмування на основі представницької передачі стану. Набір правил, що дозволяють спілкування між веб-системами за допомогою протоколу HTTP. |
| CRUD (Create, Read, Update, Delete) | Акронім, що позначає чотири базові операції, що виконуються над даними в базі даних. |
| HTTP | Це протокол для обміну даними між клієнтом і сервером. Тобто його основна задача це обмін даними між браузером (клієнтом) і віддаленим компʼютером (сервером). Обмінними проходять у формі запитів Request/Response, що буквально означає запит/відповідь. |
| WebSocket | Це протокол. який забезпечує спосіб обміну даними між браузером і сервером через постійне з’єднання. Дані можна передавати в обох напрямках у вигляді “пакетів”, без розриву з’єднання та додаткових HTTP-запитів. |
| UI (User Interface) | Користувацький інтерфейс – сукупність засобів, за допомогою яких користувач взаємодіє з програмною системою. |
| UX (User Experience) | Користувацький досвід – загальний досвід користувача при взаємодії з продуктом або послугою. |
| MVVM (Model-View-ViewModel) | Архітектурний патерн, який використовується в розробці мобільних та десктопних додатків для розділення UI, логіки та даних. |
| HTML (HyperText Markup Language) | Стандартна мова розмітки документів для створення веб-сторінок. |
| CSS (Cascading Style Sheets) | Мова стилів, яка використовується для опису представлення документа, написаного на HTML |
| JSON (JavaScript Object Notation) | Легковаговий формат обміну даними, що легко читається людиною та машиною. |
| JWT (JSON Web Token) | Компактний, URL-безпечний засіб представлення тверджень, що передаються між двома сторонами. Використовується для автентифікації та авторизації. |
| JS (JavaScript) | Високорівнева, інтерпретована мова програмування, що використовується для створення інтерактивних веб-сторінок. |
| React.js | JavaScript-бібліотека з відкритим вихідним кодом для створення інтерфейсів користувача. |
| Node.js | Середовище виконання JavaScript, що дозволяє виконувати JavaScript код поза веб-браузером. Використовується для розробки бекенд-частини. |
| Express | Мінімалістичний та гнучкий веб-фреймворк для Node.js, що надає надійний набір функцій для веб- та мобільних додатків. |
| Kotlin | Статично типізована мова програмування, яка працює на Java Virtual Machine (JVM) та є офіційно рекомендованою для розробки Android-додатків. |
| MySQL | Відкрита реляційна система управління базами даних (СУБД). |
| Firebase | Платформа для розробки мобільних і веб-додатків від Google, що надає різні сервіси, включаючи автентифікацію. |
| GPS (Global Positioning System) | Глобальна система позиціонування, що використовується для визначення місцезнаходження. |
| Nginx | Веб-сервер, який використовується як зворотний проксі-сервер і балансувальник навантаження для системи. |

## 1.4 References

1. IEEE GIDe to Software Requirements Specifications. IEEE Std 830-1998, 1998.
2. General Data Protection Regulation (GDPR). Official Journal of the European Union, 2016.
3. Google Pay API for Android. Google Developers, developers.google.com/pay/api.

## 

## 1.5 Overview

Даний документ є специфікацією для програмної системи для моніторингу стану сільськогосподарських культур структурований таким чином, щоб забезпечити послідовний та повний опис всіх аспектів системи:

* **Розділ 1 (Вступ):** Цей розділ є початковою точкою документа. Він окреслює загальний контекст системи, включаючи її призначення, окреслення обсягу функціоналу та можливих розширень, надає глосарій термінів та абревіатур, перелік посилань на джерела інформації, а також короткий огляд структури самого документа.
* **Розділ 2 (Загальний опис):** У цьому розділі система SoilScout розглядається з більш високого рівня абстракції. Він визначає перспективу продукту в цілому, описує основні функції, які надаватиме система, надає профіль типових користувачів, які взаємодіятимуть з системою, визначає загальні обмеження, що стосуються системи (наприклад, апаратні, програмні, безпекові), а також перераховує припущення та зовнішні залежності, які можуть вплинути на розробку та функціонування системи.
* **Розділ 3 (Специфічні вимоги):** Це центральний розділ SRS, який містить детальний опис всіх функціональних та нефункціональних вимог до системи. Він розбитий на підрозділи, що охоплюють:
  + **Вимоги до зовнішнього інтерфейсу:** Описує, як система взаємодіє з користувачами, апаратним та програмним забезпеченням, а також комунікаційні протоколи.
  + **Функціональні вимоги:** Детально перераховує всі необхідні функції, які повинна виконувати система, розділені за компонентами (веб-платформа, мобільний додаток, серверна частина, IoT-пристрій). Кожна вимога формулюється чітко та однозначно.
  + **Вимоги до продуктивності:** Визначає критерії швидкості, обсягу, часу відгуку та масштабованості системи.
  + **Атрибути якості програмного забезпечення:** Описує нефункціональні вимоги, такі як надійність, доступність, безпека, зручність супроводу та переносимість.
  + **Інші вимоги:** Включає додаткові вимоги, зокрема вимоги до локалізації та багатомовності.
* **Розділ 4 (Аналітичні моделі):** Цей розділ містить графічні моделі, що ілюструють ключові аспекти архітектури та взаємодії компонентів системи. Сюди входять Діаграми варіантів використання (Use Case Diagrams), що відображають взаємодію акторів з системою, Діаграми послідовності (Sequence Diagrams), що показують послідовність взаємодії об'єктів у часі, та ER-діаграма бази даних, що описує структуру зберігання даних.
* **Розділ 5 (Процес управління змінами):** Описує процедуру, яка буде використовуватися для формального внесення змін до даної специфікації, включаючи порядок подання, аналізу та затвердження змін.
* **Додатки:** Цей розділ містить додаткову інформацію, яка є важливою для розуміння системи, але не є основною частиною вимог. Приклади включають детальну специфікацію API та слайди презентації проєкту.

# 

# 

# 2. General Description

## 2.1 Product Perspective

Система SoilScout є самостійною, інноваційною та комплексною інформаційною системою, що призначена для моніторингу екологічних показників та стану ґрунту на сільськогосподарських полях.

На відміну від традиційних підходів, що часто є дорогими та повільними, SoilScout пропонує доступне та інтуїтивно зрозуміле рішення, яке надає аграріям точні та актуальні дані. Хоча SoilScout є самостійною системою, її архітектура розроблена з урахуванням можливості майбутніх інтеграцій із зовнішніми системами для розширеного аналізу даних та автоматизації процесів, наприклад, з метеостанціями для врахування погодних умов або з системами управління сільськогосподарською технікою для оптимізації внесення добрив.

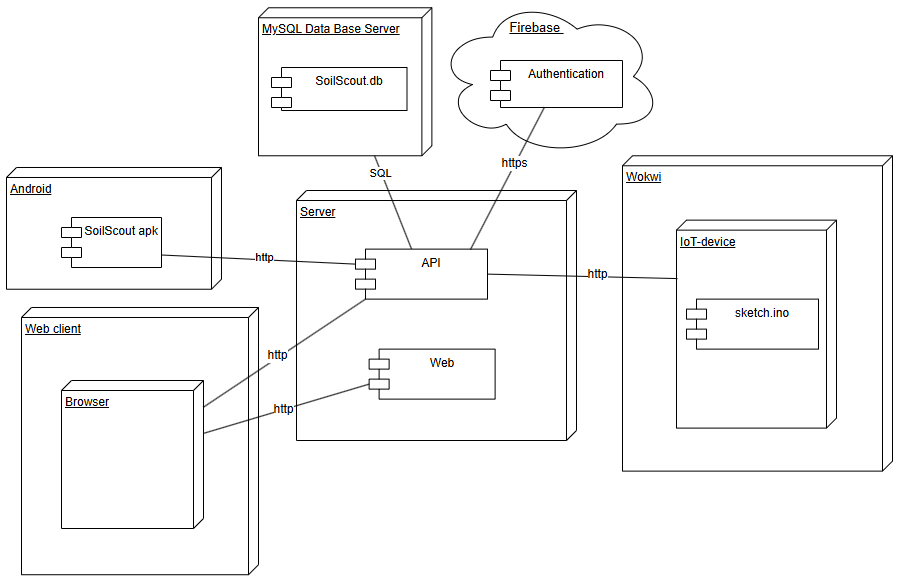


Рисунок 2.1 - Deployment Diagram

Центральним елементом системи є сервер баз даних, що працює на основі MySQL. Він відповідає за зберігання всієї інформації про користувачів, їхні поля, вимірювання, необхідні для синхронізації між різними пристроями. Ця база даних є єдиним джерелом інформації, в якому зберігаються всі дані, що потім передаються іншим компонентам системи для обробки та взаємодії з користувачем.

API сервіс виступає зв’язком між сервером баз даних, веб-сайтом і мобільним додатком. Він відповідає за обробку всіх запитів та синхронізацію даних, що забезпечує безперервність роботи на всіх платформах.

## 2.2 Product Functions

Користувачі мають можливість взаємодіяти із системою зо допомогою двох основних інтерфейсів: веб-браузер і мобільний додаток.

Веб-браузер дозволяє користувачам отримувати доступ до системи на веб-сторінках, переглядати свої поля, аналітику по них, Створювати і редагувати поля, а також надає адміністраторам можливість керувати користувачами і пристроями.

Мобільний додаток, у свою чергу, виступає в ролі додатку для проведення замірів на полях. Завдяки постійного підключення з API, користувач може бачити у мобільному застосунку отримані дані замірів у реальному часі.

## 2.3 User Characteristics

Звичайні користувачі - це можуть бути як аграрії, фермери чи звичайні фізичні особи, які реєструються та купують пристрій для моніторингу стану грунта. Вони можуть мати різний рівень технічних знань, тому інтерфейс повинен бути зручним і інтуїтивно зрозумілим.

Агрономи – це фахівці, які відповідають за планування сільськогосподарських робіт, аналіз стану ґрунту та розробку рекомендацій щодо його поліпшення. Вони мають глибокі знання в агрономії, а також досвід роботи з аналітичними програмними засобами та базами даних. Можуть мати вищі вимоги до деталізації даних та аналітичних інструментів. Їм потрібна детальна аналітика з проведення замірів на полях

Адміністратори - це користувачі, які керують системою, зокрема обліковими записами користувачів і їхніми пристроями. Адміністраторам потрібен доступ до інструментів управління за допомогою зручного веб-інтерфейсу.

Загально для всіх категорій користувачів важливими є простота реєстрації та входу, стабільна робота системи та швидкий відгук.

## 

## 2.4 General Constraints

Мобільний додаток підтримує лише Android (версія 10 і вище), а веб-додаток підтримує сучасні браузери (Chrome, Safari, Firefox, Edge).

Усі дії користувачів, включаючи купівлю книг і статус читання, повинні синхронізуватися в реальному часі між веб- та мобільними платформами.

Система повинна дотримуватися вимог GDPR щодо обробки даних користувачів.

Усі комунікації між користувачами та системою повинні бути зашифровані через HTTPS, а чутливі дані, такі як паролі, повинні бути хешовані.

## 2.5 Assumptions and Dependencies

Припущення:

* Припускається, що кінцеві користувачі матимуть доступ до стабільного інтернет-з'єднання (мобільний інтернет або Wi-Fi) для коректної роботи мобільного додатку та веб-платформи. Передача даних від IoT-пристрою через мобільний додаток також залежить від цього.
* Припускається, що спеціалізований IoT-пристрій для вимірювання параметрів ґрунту буде доступний, функціонуватиме належним чином, і його вимірювання будуть точними та надійними.
* Припускається, що користувачі мобільного додатку дадуть згоду на використання їхніх GPS-координат для відстеження їхнього фізичного місцезнаходження на полі.
* Мобільні пристрої користувачів підтримують операційні системи версії не нижче Android 10.
* Припускається, що користувачі веб-платформи будуть використовувати сучасні версії веб-браузерів, які підтримують необхідні веб-технології (HTML5, CSS3, JavaScript ES6+).
* Передбачається, що база користувачів постійно зростатиме з часом. Це припущення впливає на розробку масштабованості та резервування в архітектурі системи, але раптова кількість користувачів, яка перевищує очікування, може вимагати подальших коригувань.

Залежності:

* Функціонал автентифікації користувачів за допомогою Google, повністю залежить від безперебійної роботи сервісів Firebase. Проблеми з Firebase можуть перешкодити користувачам входити до системи або реєструватися.Синхронізація даних між веб-магазином та мобільним додатком залежить від стабільної роботи RESTful API.
* Залежність від стабільності REST API: Мобільний додаток та веб-платформа повністю залежать від стабільності та коректності роботи REST API, реалізованого на Node.js/Express. Будь-які зміни в API або його недоступність можуть призвести до некоректної роботи клієнтських додатків.
* Залежність від функціоналу GPS на мобільних пристроях: Точність та доступність GPS-координат залежить від якості GPS-модуля мобільного пристрою та умов прийому сигналу.

# 

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

Неавторизовані користувачі можуть переглядати публічні сторінки, а саме: загальну сторінку про продукт, сторінку купівлі пристроя і сторінку реєстрації і авторизації користувача. Також, їм доступний інтерфейс двома мовами – англійською та українською.

Після реєстрації користувачі зможуть за допомогою адміна прив’язати куплений пристрій до свого аккаунту і отримати доступ до кабінету. У ньому вони можуть створювати поля,

керувати ними, переглядати аналітику вимірів на них, і також переглядати свій профіль, де можна змінювати фото, ім’я та пароль.

У мобільному застосунку користувачі можуть робити все те що і на вебі, окрім перегляду аналітики, замість цього їм доступна детільна інформація про вимірювання на точках поля і сповіщення про критичні показники.

Адміністратори, окрім усього базового функціоналу звичайного користувача, мають можливість керувати користувачами і IoT пристроями.

### 3.1.2 Hardware Interfaces

Клієнтські пристрої: Система підтримує настільні комп’ютери, ноутбуки, планшети та смартфони. Основною вимогою є стабільне підключення до Інтернету для взаємодії з веб-сайтом або мобільним додатком.

Мінімальні вимоги для клієнтських пристроїв:

* Смартфони з версією операційної системи Android 10 або вище.
* Персональні комп'ютери з будь-якою сучасною операційною системою (Windows, macOS, Linux), які підтримують сучасні веб-браузери (Chrome, Firefox, Safari).

Серверні системи:

* Сервер бази даних: Сервер, на якому працює MySQL версії 9.0 або вище, повинен мати надійну апаратну конфігурацію для обробки великої кількості транзакцій.

### 

### 3.1.3 Software Interfaces

Система використовує кілька програмних інтерфейсів для забезпечення роботи:

* Операційні системи:
  + Для клієнтських пристроїв. Мобільні операційні системи Android (версія 10 і вище). Для настільних користувачів — будь-яка операційна система, яка підтримує сучасні веб-браузери.
  + Для серверів. Лінукс.
* Інтерфейс бази даних. Система баз даних використовує MySQL версії 8.0 і вище. Для взаємодії з базою даних використовуються стандартні SQL-запити.
* API інтерфейси. Використовується RESTful API. Вся система побудована на взаємодії між веб-додатком, мобільним додатком і сервером баз даних через API.
* Використовуються сторонні сервіси, такі як: Firebase і Google Maps API.

### 3.1.4 Communications Interfaces

Комунікаційні інтерфейси забезпечують зв'язок між клієнтськими пристроями, сервером та сторонніми сервісами:

* Для обміну даними між клієнтськими пристроями (веб-браузери та мобільні додатки) і сервером використовується HTTP.
* Комунікація між веб-додатком, мобільним додатком і сервером відбувається за допомогою RESTful API, яка забезпечує передачу даних між клієнтськими інтерфейсами та базою даних. Всі API запити використовують стандартні методи HTTP (GET, POST, PUT, DELETE) для виконання різних операцій, таких як завантаження списку полів та синхронізації даних.

## 

## 3.2 Functional Requirements

### 3.2.1 Реєстрація користувача

#### 3.2.1.1 Introduction

У цьому розділі описано процес реєстрації користувачів у системі. Користувачі зможуть зареєструватися, надавши адресу електронної пошти, пароль та ім'я користувача, або за допомогою Google аккаунту. Бекенд буде обробляти аутентифікацію та валідувати введені дані.

#### 3.2.1.2 Inputs

* Адреса електронної пошти, надана користувачем (має бути унікальною).
* Пароль, встановлений користувачем (відповідно до вимог системи до паролів).
* Нікнейм користувача який буде відображатися в його профілі.

#### 3.2.1.3 Processing

* Отримавши вхідні дані, система надсилає адресу електронної пошти, пароль та нікнейм.
* Сервер перевіряє, чи дійсна електронна адреса і чи не використовується вона раніше.
* Якщо електронна пошта та пароль відповідають критеріям перевірки, Створюємо новий обліковий запис користувача у своїй системі аутентифікації.
* База даних генерує унікальний ідентифікатор користувача для нового облікового запису.
* Інформація про користувача зберігається в базі даних MySQL системи.

#### 3.2.1.4 Outputs

* У разі успішної реєстрації, ідентифікатор користувача, адреса електронної пошти та інша необхідна інформація зберігається у базі даних. База даних MySQL зберігає інформацію про користувача.
* Система надає доступ до основної функціональності (залежно від типу аккаунту клієнт/адміністратор).

#### 3.2.1.5 Error Handling

* кщо формат електронної пошти невірний, система відобразить повідомлення про помилку і запропонує користувачеві ввести коректну адресу.
* Якщо електронна пошта вже зареєстрована, система запропонує користувачеві ввести іншу пошту, увійти або відновити пароль.
* Якщо пароль не відповідає вимогам безпеки системи (довжина, складність тощо), система виведе повідомлення про помилку і запропонує користувачу вибрати більш надійний пароль.
* Якщо є проблеми з підключенням, система повідомить користувача про необхідність повторити спробу пізніше.

### 3.2.2 Авторизація користувача

#### 3.2.2.1 Introduction

Цей розділ описує процес авторизації користувача в системі. Користувачі зможуть увійти, використовуючи свою електронну пошту та пароль, або за допомогою прив'язаного Google-акаунту.

#### 3.2.2.2 Inputs

Для авторизації користувач повинен надати такі дані:

* Email: Адреса електронної пошти, зареєстрована під час реєстрації.
* Пароль: Пароль, вказаний під час реєстрації.

#### 3.2.2.3 Processing

* Користувач вводить свою електронну пошту та пароль у форму авторизації.
* Система надсилає введені дані до бекенду для перевірки.
* Бекенд звіряє облікові дані з наявними у своїй базі даних (або взаємодіє з Firebase для Google-авторизації).
* Якщо дані правильні, бекенд повертає успішну аутентифікацію та токен доступу.

#### 3.2.2.4 Outputs

* У разі успішного входу, користувач отримує доступ до свого профілю та функціоналу системи.
* Система отримує ідентифікатор користувача та токен доступу для подальшої взаємодії.

#### 3.2.2.5 Error Handling

* Якщо користувач ввів неправильну електронну пошту або пароль, система виводить повідомлення про помилку і пропонує спробувати знову.
* Якщо є проблеми з підключенням до сервера або Firebase, система повідомляє користувача про неможливість авторизації в даний момент.

### 3.2.3 Перегляд сторінки профілю

#### 3.2.3.1 Introduction

Цей розділ описує функціонал сторінки профілю користувача в мобільному додатку та на веб-сайті, де відображається його основна інформація.

#### 3.2.3.2 Inputs

* Для перегляду сторінки профілю використовується ідентифікатор користувача, отриманий під час авторизації.

#### 3.2.3.3 Processing

* Система отримує ідентифікатор поточного авторизованого користувача.
* Запитує дані профілю (нікнейм, email, фото профілю) з бекенду.
* Бекенд отримує дані з бази даних та повертає їх.

#### 3.2.3.4 Outputs

* На сторінці профілю відображається нікнейм, email та фотографія профілю користувача.
* На цій сторінці у мобільному додатку відображається кнопка виходу з профілю, що дозволяє користувачу вийти зі свого акаунту.

#### 3.2.3.5 Error Handling

* Якщо дані профілю не можуть бути завантажені, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість завантаження профілю.

**3.2.4 Редагування профілю**

3.2.4.1 Introduction  
Цей розділ описує функціонал редагування інформації профілю користувача, включаючи зміну фото, імені та паролю, пароль можна змінювати лише якщо реєстрація була локальною. Цей функціонал доступний як на веб-платформі, так і в мобільному додатку.

3.2.4.2 Inputs

* Нове фото профілю (опціонально).
* Нове ім'я користувача (опціонально).
* Старий пароль (опціонально).
* Новий пароль (обов’язково якщо поле старий пароль непусте).
* Підтвердження нового паролю (обов’язково для підтвердження нового паролю).

3.2.4.3 Processing

* Користувач вносить зміни у відповідні поля форми редагування профілю.
* Система валідує введені дані (наприклад, відповідність паролів, вимоги до нового пароля).
* Валідовані дані надсилаються до бекенду.
* Бекенд оновлює інформацію про користувача в базі даних.
* Для зміни пароля, бекенд перевіряє старий пароль перед оновленням.

3.2.4.4 Outputs

* Інформація профілю користувача (фото, ім'я, пароль) оновлюється.
* Користувач отримує візуальне підтвердження успішного оновлення.

3.2.4.5 Error Handling

* Якщо введені дані недійсні (наприклад, паролі не співпадають, або старий пароль невірний), система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера або базою даних, система повідомляє про неможливість оновлення профілю.

### 3.2.5 Вихід з профілю

3.2.5.1 Introduction  
Цей розділ описує функціонал виходу користувача з облікового запису в мобільному додатку та на веб-сайті. Функція виходу дозволяє користувачу безпечно завершити сеанс роботи в системі, припиняючи доступ до особистих даних до наступного входу.

3.2.5.2 Inputs

* Токен авторизації.

3.2.5.3 Processing

* Користувач натискає кнопку "Вийти".
* Система анулює сесію користувача і очищає його localStorage.
* Перенаправляє користувача на сторінку логіну.

3.2.5.4 Outputs

* Користувач виходить з системи та перенаправляється на сторінку входу.

#### 

#### 3.2.5.5 Error Handling

* Якщо виникають проблеми з анулюванням сесії, система може повідомити користувача про помилку.

### 3.2.6 Перегляд списку полів

#### 3.2.6.1 Introduction

Цей розділ описує функціонал перегляду списку полів, які належать користувачеві, на веб-платформі та в мобільному додатку.

#### 3.2.6.2 Inputs

* Ідентифікатор авторизованого користувача.

#### 3.2.6.3 Processing

* Система надсилає запит до бекенду для отримання списку полів, пов'язаних з поточним користувачем.
* Бекенд отримує дані про поля з бази даних та повертає їх.
* Фронтенд відображає отриманий список полів.

#### 3.2.6.4 Outputs

* На веб-платформі список полів постійно відображається.
* На мобільній платформі список полів відображається на окремій сторінці "Дашборд".

#### 3.2.6.5 Error Handling

* Якщо список полів не може бути завантажений, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість завантаження списку полів.

### 3.2.7 Пошук полів

#### 3.2.7.1 Introduction

Цей розділ описує функціонал пошуку полів на клієнтській частині (веб-платформа та мобільний додаток).

#### 3.2.7.2 Inputs

* Пошуковий запит (назва поля)

#### 3.2.7.3 Processing

* Користувач вводить пошуковий запит у поле пошуку.
* Клієнтська частина фільтрує відображений список полів на основі введеного запиту.

#### 3.2.7.4 Outputs

* Відображається відфільтрований список полів, що відповідають пошуковому запиту.

#### 3.2.7.5 Error Handling

* Якщо полів, що відповідають запиту, не знайдено, система відображає повідомлення про це.

### 3.2.8 Сортування полів

#### 3.2.8.1 Introduction

Цей розділ описує функціонал сортування списку полів на клієнтській частині (веб-платформа та мобільний додаток) за назвою, площею та датою створення.

#### 3.2.8.2 Inputs

* Критерій сортування (назва, площа, дата створення).
* Напрямок сортування (за зростанням, за спаданням).

#### 3.2.8.3 Processing

* Користувач обирає критерій та напрямок сортування.
* Клієнтська частина сортує відображений список полів відповідно до обраних параметрів.

#### 3.2.8.4 Outputs

* Відображається відсортований список полів.

#### 3.2.8.5 Error Handling

* Якщо немає жодного поля який би підпадав під параметри сортування, то виведеться повідомлення що полів не знайдено.

### 3.2.9 Фільтрація полів за активністю

#### 3.2.9.1 Introduction

Цей розділ описує функціонал фільтрації списку полів за їхнім активним статусом (чи було поле останнє відкрите) на клієнтській частині (веб-платформа та мобільний додаток).

#### 3.2.9.2 Inputs

* Стан активності (активне/неактивне).

#### 3.2.9.3 Processing

* Користувач обирає критерій фільтрації за активністю.
* Клієнтська частина фільтрує відображений список полів на основі їхнього статусу.

#### 3.2.9.4 Outputs

* Відображається відфільтрований список полів, що відповідають обраному статусу активності.

#### 3.2.9.5 Error Handling

* Якщо немає жодного поля який би підпадав під фільтр, то виведеться повідомлення що полів не знайдено.

### 3.2.10 Створення поля

#### 3.2.10.1 Introduction

Цей розділ описує функціонал створення нового поля, включаючи введення назви, площі та координат. На веб-платформі також доступний інтерактивний спосіб введення координат за допомогою Google Maps.

#### 3.2.10.2 Inputs

* ID користувача
* Назва поля (обов'язково).
* Площа поля (числове значення).
* Координати поля (набір географічних точок, що утворюють полігон).
* Взаємодія з Google Maps для малювання полігону (для веб-платформи).

#### 3.2.10.3 Processing

* Користувач вводить назву та площу поля.

На веб-платформі:

* Користувач може вручну ввести координати.
* Або користувач може використовувати інтерактивну карту Google для малювання полігону, при цьому площа та координати автоматично заповнюються.

На мобільній платформі:

* + Користувач вручну вводить площу та координати.
* Після введення всіх даних, вони надсилаються на бекенд.
* Бекенд валідує дані та зберігає інформацію про нове поле у базі даних, прив'язуючи його до поточного користувача.

#### 3.2.10.4 Outputs

* Нове поле успішно створюється та додається до списку полів користувача.
* Користувач отримує візуальне підтвердження успішного створення.

#### 3.2.10.5 Error Handling

* Якщо назва поля відсутня або дані недійсні, система відображає повідомлення про помилку валідації.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість створення поля.

### 3.2.11 Редагування поля

#### 3.2.11.1 Introduction

Цей розділ описує функціонал редагування існуючого поля. Ця сторінка є тією ж, що й сторінка створення поля, але з попередньо заповненою інформацією про поле, яке редагується.

#### 3.2.11.2 Inputs

* Ідентифікатор поля, що редагується.
* Оновлена назва поля (опціонально).
* Оновлена площа поля (опціонально).
* Оновлені координати поля (опціонально).
* Взаємодія з Google Maps для зміни полігону (для веб-платформи).

#### 3.2.11.3 Processing

* Користувач обирає поле для редагування.
* Система завантажує існуючі дані поля у форму.
* Користувач вносить зміни до відповідних полів.
* Система валідує введені дані.
* Оновлені дані надсилаються до бекенду.
* Бекенд оновлює інформацію про поле у базі даних.

#### 3.2.11.4 Outputs

* Інформація про поле успішно оновлюється.
* Користувач отримує візуальне підтвердження успішного оновлення.

#### 3.2.11.5 Error Handling

* Якщо введені дані недійсні, система відображає повідомлення про помилку валідації.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість оновлення поля.

### 3.2.12 Видалення поля

#### 3.2.12.1 Introduction

#### Цей розділ описує функціонал видалення існуючого поля.

#### 3.2.12.2 Inputs

* Ідентифікатор поля, що підлягає видаленню.

#### 3.2.12.3 Processing

#### Користувач ініціює видалення поля за допомогою кнопки на сторінці редагування поля.

#### Система запитує підтвердження видалення.

#### Після підтвердження, запит на видалення надсилається до бекенду.

#### Бекенд видаляє поле та всі пов'язані з ним дані (точки вимірювань, історію вимірювань) з бази даних.

#### 3.2.12.4 Outputs

#### Поле успішно видаляється зі списку полів користувача.

#### Користувач отримує візуальне підтвердження успішного видалення.

#### 3.2.12.5 Error Handling

* Якщо виникають проблеми з підключенням до сервера або базою даних, система повідомляє про неможливість видалення поля.

### 3.2.13 Перегляд сторінки поля

#### 3.2.13.1 Introduction

Цей розділ описує функціонал перегляду детальної інформації про конкретне поле, включаючи назву, площу, дату створення та список точок вимірювань.

#### 3.2.13.2 Inputs

* Ідентифікатор поля.

#### 3.2.13.3 Processing

* Користувач обирає поле для перегляду.
* Система надсилає запит до бекенду для отримання детальної інформації про поле та пов'язані з ним точки вимірювань.

Бекенд отримує дані з бази даних та повертає їх.

#### 3.2.13.4 Outputs

На веб-платформі:

* Відображається назва поля, площа, дата створення.
* Список точок вимірювань з їх номером, координатами, поточним статусом (активна/неактивна) та кнопками активації/деактивації.

На мобільній платформі:

* Відображається назва поля, площа, дата створення.
* Інтерактивна карта Google з відображенням точок вимірювань.

#### 3.2.13.5 Error Handling

* Якщо інформація про поле не може бути завантажена, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість завантаження сторінки поля.

### 3.2.14 Передача даних з IoT-пристрою

#### 3.2.14.1 Introduction

Цей розділ описує процес передачі даних про вологість, температуру та кислотність ґрунту з IoT-пристрою до системи.

#### 3.2.14.2 Inputs

* Унікальний ідентифікатор IoT-пристрою.

Дані вимірювань з датчиків:

* Вологість ґрунту (%).
* Температура ґрунту (°C).
* Кислотність ґрунту (pH).
* Час та дата вимірювання.

#### 3.2.14.3 Processing

* IoT-пристрій підключається до мережі.
* Пристрій відправляє HTTP-запит на сервер, передаючи свій унікальний ідентифікатор.
* Сервер визначає активне поле користувача, до якого прив'язаний цей пристрій, та активну точку на цьому полі.
* IoT-пристрій формує JSON-об'єкт з отриманими даними вимірювання від сенсорів.
* Сформований JSON-об'єкт передається на сервер для збереження на відповідній активній точці.
* Бекенд валідує отримані дані та зберігає їх у базі даних, прив'язуючи до відповідної точки вимірювання.

#### 3.2.14.4 Outputs

* Дані вимірювань успішно зберігаються в базі даних та відображаються для відповідної точки.
* Система може ініціювати генерацію сповіщень у разі критичних показників.

#### 3.2.14.5 Error Handling

* Якщо ідентифікатор пристрою недійсний або пристрій не прив'язаний до активного поля/точки, система відхиляє дані.
* Якщо формат даних вимірювань невірний, система повідомляє про помилку валідації.
* Якщо виникають проблеми з підключенням до сервера, дані не зберігаються.

### 3.2.15 Перегляд деталей точки (мобільний додаток)

#### 3.2.15.1 Introduction

Цей розділ описує функціонал перегляду детальної інформації про конкретну точку вимірювання в мобільному додатку.

#### 3.2.15.2 Inputs

* Ідентифікатор точки вимірювання.

#### 3.2.15.3 Processing

* Користувач обирає точку на карті або зі списку.
* Мобільний додаток надсилає запит до бекенду для отримання детальної інформації про точку та останні вимірювання.
* Бекенд отримує дані з бази даних та повертає їх.

#### 3.2.15.4 Outputs

* Відображається номер точки на полі, її координати та статус (активна/неактивна).
* Відображаються останні дані вимірювання (вологість, температура, pH), що надійшли з IoT-пристрою для цієї точки.
* Доступні кнопки активації/деактивації точки.

#### 3.2.15.5 Error Handling

* Якщо інформація про точку не може бути завантажена, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість завантаження деталей точки.

### 3.2.16 Отримання сповіщень (мобільний додаток)

#### 3.2.16.1 Introduction

Цей розділ описує процес отримання сповіщень на Android-застосунок, якщо хоча б один з отриманих показників ґрунту (вологість, температура, кислотність) виходить за межі норми.

#### 3.2.16.2 Inputs

* Дані вимірювань з IoT-пристрою.
* Встановлені порогові значення для кожного показника ґрунту.

#### 3.2.16.3 Processing

* Андройд аналізує вхідні дані з IoT-пристроїв.
* Порівнює кожен показник (вологість, температура, pH) з встановленими для відповідного поля/точки пороговими значеннями.
* Якщо будь-який показник виходить за межі норми (занизький або зависокий), бекенд генерує сповіщення.
* Сповіщення формується у форматі: така-то точка має такий-то занизький/зависокий показник на такому-то полі.
* Сповіщення надсилається за допомогою push-повідомлень.

#### 3.2.16.4 Outputs

* Користувач отримує push-повідомлення на свій Android-застосунок про критичний показник ґрунту.
* Сповіщення додається до списку всіх сповіщень у застосунку.

#### 3.2.16.5 Error Handling

* Якщо система push-повідомлень недоступна або виникають проблеми з доставкою, сповіщення може бути відкладено або не доставлено.
* Якщо налаштування порогових значень не визначені, сповіщення не генеруються.

### 3.2.17 Перегляд усіх сповіщень (мобільний додаток)

#### 3.2.17.1 Introduction

Цей розділ описує функціонал перегляду списку всіх отриманих сповіщень на сторінці "Notifications" у мобільному додатку.

#### 3.2.17.2 Inputs

* Ідентифікатор авторизованого користувача.

#### 3.2.17.3 Processing

* Користувач переходить на сторінку "Notifications" у мобільному додатку.
* Мобільний додаток надсилає запит до бекенду для отримання списку всіх сповіщень, пов'язаних з цим користувачем.
* Бекенд отримує дані про сповіщення з бази даних та повертає їх.

#### 3.2.17.4 Outputs

* На сторінці "Notifications" відображається список усіх сповіщень.

#### 3.2.17.5 Error Handling

* Якщо список сповіщень не може бути завантажений, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість завантаження сповіщень.

### 3.2.18 Видалення всіх сповіщень (мобільний додаток)

#### 3.2.18.1 Introduction

Цей розділ описує функціонал видалення всіх сповіщень зі списку на сторінці "Notifications" у мобільному додатку.

#### 3.2.18.2 Inputs

* Ідентифікатор авторизованого користувача.

#### 3.2.18.3 Processing

* Користувач натискає кнопку "Видалити всі сповіщення".
* Мобільний додаток надсилає запит до бекенду для видалення всіх сповіщень, пов'язаних з цим користувачем.
* Бекенд видаляє всі відповідні записи з бази даних.

#### 3.2.18.4 Outputs

* Список сповіщень очищується.
* Користувач отримує візуальне підтвердження успішного видалення.

#### 3.2.18.5 Error Handling

* Якщо виникають проблеми з підключенням до сервера або базою даних, система повідомляє про неможливість видалення сповіщень.

### 3.2.19 Перегляд аналітики поля (веб-платформа)

#### 3.2.19.1 Introduction

Цей розділ описує функціонал перегляду детальної аналітики полів на веб-платформі, включаючи графіки показників ґрунту за різні періоди.

#### 3.2.19.2 Inputs

* Ідентифікатор поля для аналітики.
* Вибраний період аналізу (7 днів, 6 місяців, 1 рік).

#### 3.2.19.3 Processing

* Користувач обирає поле та період для аналітики.
* Система надсилає запит до бекенду для отримання даних вимірювань за вказаний період для обраного поля.
* Бекенд збирає дані з бази даних та повертає їх.
* Фронтенд використовує Chart.js для візуалізації даних у вигляді чотирьох графіків: загальний, вологості, температури та кислотності ґрунту.

#### 3.2.19.4 Outputs

* Відображається сторінка аналітики поля з чотирма інтерактивними графіками, що показують динаміку змін обраних показників ґрунту за вибраний період.

#### 3.2.19.5 Error Handling

* Якщо дані за обраний період відсутні, система відображає повідомлення про відсутність даних.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість завантаження аналітичних даних.

### 3.2.20 Керування користувачами (для адміністраторів)

3.2.20.1 Introduction

Цей розділ описує функціонал, доступний адміністраторам системи, для пошуку, перегляду інформації та видалення облікових записів користувачів.

3.2.20.2 Inputs

* Пошуковий запит (наприклад, email або нікнейм користувача).
* Ідентифікатор користувача для перегляду/видалення.

3.2.20.3 Processing

* Система надсилає запит до бекенду для отримання інформації про користувачів.
* Бекенд перевіряє права адміністратора, і повертає дані з бази даних.
* Адміністратор вводить пошуковий запит або обирає користувача зі списку.
* Система відфільтровує список користувачів якщо був ініціалізований пошук.
* Для видалення: адміністратор ініціює видалення за допомогою кнопки на вебі, бекенд видаляє обліковий запис.

3.2.20.4 Outputs

* Відображається список користувачів з можливістю пошуку.
* Можливість перегляду детальної інформації про кожного користувача.
* Можливість видалення користувача зі списку.

3.2.20.5 Error Handling

* Якщо адміністратор не має достатніх прав, система відмовляє в доступі.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість виконання операції.

### 3.2.21 Керування девайсами (для адміністраторів)

#### 3.2.21.1 Introduction

Цей розділ описує функціонал, доступний адміністраторам системи, для створення, редагування та видалення записів про IoT-пристрої.

#### 3.2.21.2 Inputs

* Дані нового пристрою.
* Ідентифікатор існуючого пристрою для редагування або видалення.
* Оновлені дані пристрою.

#### 3.2.21.3 Processing

* Система надсилає запит до бекенду для отримання інформації про девайси.
* Бекенд перевіряє права адміністратора, і повертає дані з бази даних.
* Система відображає список існуючих пристроїв.
* Адміністратор ініціює дію (створення/редагування/видалення) через адміністративний інтерфейс і вводить дані у форму для введення/редагування даних пристрою.
* Введені дані надсилаються на бекенд.
* Бекенд перевіряє права адміністратора та виконує відповідну операцію (створення/оновлення/видалення) у базі даних.

#### 3.2.21.4 Outputs

* Оновлений список IoT-пристроїв відображається в адміністративній панелі.
* Адміністратор отримує візуальне підтвердження успішної операції.

#### 3.2.21.5 Error Handling

* Якщо дані пристрою недійсні або адміністратор не має достатніх прав, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість виконання операції.

### 3.2.22 Керування сенсорами (для адміністраторів)

#### 3.2.22.1 Introduction

Цей розділ описує функціонал, доступний адміністраторам системи, для створення, редагування та видалення записів про сенсори, які надають виміри в IoT-пристроях.

#### 3.2.22.2 Inputs

* Дані нового сенсора (наприклад, тип сенсора, діапазон вимірювань, одиниці вимірювання).
* Ідентифікатор існуючого сенсора для редагування або видалення.
* Оновлені дані сенсора.

#### 3.2.22.3 Processing

* Система надсилає запит до бекенду для отримання інформації про сенсори.
* Бекенд перевіряє права адміністратора, і повертає дані з бази даних.
* Система відображає список існуючих сенсорів.
* Адміністратор ініціює дію (створення/редагування/видалення) сенсора через адміністративний інтерфейс і вводить дані у форму для введення/редагування даних сенсора.
* Введені дані надсилаються на бекенд.
* Бекенд перевіряє права адміністратора та виконує відповідну операцію у базі даних.

#### 3.2.22.4 Outputs

* Оновлений список сенсорів відображається в адміністративній панелі.
* Адміністратор отримує візуальне підтвердження успішної операції.

#### 3.2.22.5 Error Handling

* Якщо дані сенсора недійсні або адміністратор не має достатніх прав, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість виконання операції.

### 3.2.23 Керування пристроями користувачів (для адміністраторів)

#### 3.2.23.1 Introduction

Цей розділ описує функціонал, доступний адміністраторам системи, для створення та видалення зв'язку між IoT-пристроями та користувачами.

#### 3.2.23.2 Inputs

* Ідентифікатор користувача.
* Ідентифікатор IoT-пристрою для прив'язки.

#### 3.2.23.3 Processing

* Система надсилає запит до бекенду для отримання інформації про зв'язок пристроїв з користувачами.
* Бекенд перевіряє права адміністратора, і повертає дані з бази даних.
* Система відображає список користувачів з їхніми пристроями.
* Адміністратор обирає користувача та IoT-пристрій для прив'язки або видалення зв'язку.
* Надсилається відповідний запит прив’язки чи відв’язки до бекенду.
* Бекенд перевіряє права адміністратора та створює/видаляє запис у базі даних, що пов'язує користувача з пристроєм.

#### 3.2.23.4 Outputs

* Створюється або видаляється зв'язок між користувачем та IoT-пристроєм.
* Адміністратор отримує візуальне підтвердження успішної операції.

#### 3.2.23.5 Error Handling

* Якщо дані недійсні або адміністратор не має достатніх прав, система відображає повідомлення про помилку.
* Якщо виникають проблеми з підключенням до сервера, система повідомляє про неможливість виконання операції.

## 3.3 Non-Functional Requirements

### 3.3.1 Performance

* Час відгуку: Система повинна забезпечувати час відгуку на запити користувача не більше 2 секунд для більшості операцій (перегляд списку полів, перегляд профілю, відображення аналітики).
* Масштабованість: Система повинна підтримувати одночасну роботу до 1000 активних користувачів без помітного зниження продуктивності.
* Обробка даних: Система повинна обробляти та зберігати дані з IoT-пристроїв у режимі реального часу, з мінімальною затримкою (не більше 5 секунд від моменту отримання даних пристроєм до їх відображення в системі).

### 3.3.2 Reliability

* На початковій стадії система має забезпечувати >500 годин MTBF. У проєкті багато залежностей, наприклад, від MySQL, API і синхронізації, що збільшує ймовірність збоїв у системі. У подальшому, цей показник можна збільшити до 1000 годин за умови правильного налаштування і тестування.
* Система повинна коректно обробляти помилки, надаючи користувачам інформативні повідомлення.
* Час відновлення після непередбачених збоїв не повинен перевищувати 1 годину.

### 3.3.4 Security

* Аутентифікація та авторизація: Система повинна забезпечувати надійну аутентифікацію користувачів за допомогою електронної пошти/паролю та Google-акаунту, використовуючи токени (JWT) для авторизації.
* Захист даних: Всі передані дані між клієнтом (веб, мобільний додаток) та сервером повинні бути зашифровані за допомогою HTTPS.
* Захист від несанкціонованого доступу: Система повинна запобігати несанкціонованому доступу до даних та функціоналу шляхом перевірки прав доступу для кожної операції.
* Захист від XSS/CSRF: Веб-платформа повинна бути захищена від відомих вразливостей, таких як XSS (Cross-Site Scripting) та CSRF (Cross-Site Request Forgery).

### 3.3.6 Portability

* Додаток має підтримувати операційну систему Android версії не нижче 10.
* Веб-інтерфейс повинен коректно працювати в сучасних браузерах, таких як Chrome, Firefox, Safari, Opera, Microsoft Edge.

## 3.4 Classes / Objects

### 3.4.1 field\_measurements

#### 3.4.1.1 Attributes

* id (int): Унікальний ідентифікатор вимірювання поля.
* field\_id (int): Ідентифікатор поля, до якого належить вимірювання (зовнішній ключ до таблиці fields).
* sensor\_type (varchar(50)): Тип сенсора (наприклад, 'temperature', 'soil\_moisture', 'acidity').
* avg\_value (float): Середнє значення вимірювання.
* min\_value (float): Мінімальне значення вимірювання.
* max\_value (float): Максимальне значення вимірювання.
* measurement\_date (date): Дата проведення вимірювання.
* created\_at (datetime): Дата та час створення запису.

3.4.1.2 Functions

* Отримання вимірювань поля за певний період (7 днів, 6 місяців, 1 рік)
* Перегляд аналітики поля (використання для побудови графіків)

### 3.4.2 fields

#### 3.4.2.1 Attributes

* id (int): Унікальний ідентифікатор поля.
* user\_id (int): Ідентифікатор користувача, якому належить поле (зовнішній ключ до таблиці users).
* name (varchar(255)): Назва поля.
* area (float): Площа поля.
* geo\_zone (json): Географічні координати поля (полігон).
* selected (tinyint(1)): Прапорець, що вказує, чи є поле активним/вибраним (0 - неактивне, 1 - активне).
* created\_at (datetime): Дата та час створення запису.

#### 3.4.2.2 Functions

* Створення поля
* Редагування поля
* Видалення поля
* Перегляд списку полів
* Пошук полів
* Сортування полів
* Фільтрація полів за активністю
* Перегляд сторінки поля
* Перегляд аналітики поля

### 3.4.3 iot\_devices

#### 3.4.3.1 Attributes

* id (int): Унікальний ідентифікатор IoT-пристрою.
* device\_name (varchar(255)): Назва пристрою (унікальна).
* created\_at (timestamp): Дата та час створення запису.
* updated\_at (timestamp): Дата та час останнього оновлення запису.

#### 3.4.3.2 Functions

* Керування девайсами (для адміністраторів)
* Передача даних з IoT-пристрою

### 3.4.4 measurement\_points

#### 3.4.4.1 Attributes

* id (int): Унікальний ідентифікатор точки вимірювання.
* field\_id (int): Ідентифікатор поля, до якого належить точка (зовнішній ключ до таблиці fields).
* latitude (double): Широта точки вимірювання.
* longitude (double): Довгота точки вимірювання.
* point\_order (int): Порядковий номер точки на полі.
* active (tinyint): Прапорець, що вказує, чи є точка активною (0 - неактивна, 1 - активна).

#### 3.4.4.2 Functions

* Додавання точок вимірювань
* Перегляд сторінки поля (відображення точок)
* Перегляд деталей точки (мобільний додаток)
* Передача даних з IoT-пристрою (зв'язок з точкою)

### 3.4.5 measurements

#### 3.4.5.1 Attributes

* id (int): Унікальний ідентифікатор вимірювання.
* point\_id (int): Ідентифікатор точки вимірювання, до якої належить вимірювання (зовнішній ключ до таблиці measurement\_points).
* sensor\_id (int): Ідентифікатор сенсора, що здійснив вимірювання (зовнішній ключ до таблиці sensors).
* value (float): Значення вимірювання.
* timestamp (datetime): Дата та час проведення вимірювання.

#### 3.4.5.2 Functions

* Передача даних з IoT-пристрою (збереження вимірювань)
* Перегляд аналітики поля (отримання даних для графіків)
* Отримання сповіщень (аналіз значень)
* Перегляд деталей точки (відображення останніх вимірювань)

### 3.4.6 refresh\_sessions

#### 3.4.6.1 Attributes

* id (int): Унікальний ідентифікатор сесії оновлення токена.
* user\_id (int): Ідентифікатор користувача, до якого належить сесія (зовнішній ключ до таблиці users).
* refresh\_token (char(36)): Токен оновлення сесії.
* expires\_in (bigint): Термін дії токена оновлення.
* created\_at (timestamp): Дата та час створення запису.

#### 3.4.6.2 Functions

* Авторизація користувача (керування сесіями)
* Вихід з профілю (видалення сесій)

### 3.4.7 sensors

#### 3.4.7.1 Attributes

* id (int): Унікальний ідентифікатор сенсора.
* type (enum('temperature','soil\_moisture','acidity')): Тип сенсора.
* unit (varchar(25)): Одиниці вимірювання.
* status (enum('active','inactive','maintenance','calibrating','error','retired')): Статус сенсора.
* radius (double): Радіус дії сенсора (якщо застосовно).

#### 3.4.7.2 Functions

* Керування сенсорами (для адміністраторів)
* Передача даних з IoT-пристрою (визначення типу сенсора)

### 3.4.8 user\_iot\_devices

#### 3.4.8.1 Attributes

* id (int): Унікальний ідентифікатор зв'язку користувача з IoT-пристроєм.
* user\_id (int): Ідентифікатор користувача (зовнішній ключ до таблиці users).
* iot\_device\_id (int): Ідентифікатор IoT-пристрою (зовнішній ключ до таблиці iot\_devices).
* assigned\_at (timestamp): Дата та час прив'язки пристрою до користувача.

#### 3.4.8.2 Functions

* Керування пристроями користувача (для адміністраторів)
* Передача даних з IoT-пристрою (визначення приналежності пристрою користувачеві)

### 3.4.9 users

#### 3.4.9.1 Attributes

* id (int): Унікальний ідентифікатор користувача.
* uid (varchar(255)): Унікальний ідентифікатор користувача (для Firebase/Google).
* email (varchar(255)): Адреса електронної пошти (унікальна).
* password (varchar(255)): Хеш пароля (для локальної реєстрації).
* name (varchar(255)): Нікнейм користувача.
* is\_admin (tinyint(1)): Прапорець, що вказує, чи є користувач адміністратором (0 - ні, 1 - так).
* profile\_picture\_url (text): URL-адреса фотографії профілю.

#### 3.4.9.2 Functions

* Реєстрація користувача
* Авторизація користувача
* Перегляд сторінки профілю
* Редагування профілю
* Вихід з профілю
* Перегляд списку полів (власник)
* Керування користувачами (для адміністраторів)
* Керування пристроями користувача (для адміністраторів)

## 3.5 Inverse Requirements

* Мобільний застосунок не повинен надавати детальної аналітики полів. (Ця функціональність, включаючи графіки та статистику, доступна лише через веб-версію).
* Система не має надавати користувачеві персоналізовані рекомендації щодо дій з полем на основі зібраних даних. (Наприклад, система не буде автоматично рекомендувати "полити поле" або "внести добрива", а лише надавати сирі дані та сповіщення про вихід за межі норми).
* Користувач не має можливості завантажувати власні файли даних вимірювань для інтеграції в систему. (Всі дані надходять виключно з підключених IoT-пристроїв).
* Мобільний додаток не повинен підтримувати операційну систему iOS; він призначений тільки для Android. (Підтримка iOS може бути розглянута у майбутніх релізах).
* Система не повинна здійснювати автоматичне керування IoT-пристроями. (Користувач взаємодіє з пристроєм та ініціює вимірювання вручну, а система лише отримує дані від пристрою).

## 3.6 Design Constraints

* Відповідність GDPR: Система повинна відповідати актуальним вимогам GDPR (General Data Protection Regulation) в області обробки персональних даних користувачів. Це включає принципи згоди на обробку даних, право на забуття, захист даних за замовчуванням та інші аспекти приватності.
* Можливість зміни СУБД та зовнішніх компонентів: Архітектура системи повинна передбачати можливість заміни поточної СУБД (MySQL), використовуваних бібліотек (наприклад, OkHttp3, Chart.js, React.js) або інших зовнішніх компонентів у разі припинення їх підтримки, зміни ліцензійної політики або виявлення критичних вразливостей, які можуть вплинути на роботу системи.
* Можливість переходу на інші платформи для аутентифікації: Система повинна бути спроєктована таким чином, щоб у разі зміни політики роботи або припинення підтримки поточної платформи для аутентифікації (Firebase), був можливий відносно безболісний перехід на альтернативну систему аутентифікації без значних перероблень основної логіки.
* Використання хмарної платформи Microsoft Azure: Розгортання та функціонування системи (веб-платформа, база даних, бекенд) повинно здійснюватися на хмарній платформі Microsoft Azure, з урахуванням її можливостей та обмежень.
* Технологічний стек: Дотримання визначеного технологічного стеку: React.js з Tailwind CSS та JavaScript для фронтенду, Node.js з Express для бекенду, Kotlin/Java+XML для Android-додатку.
* Мінімальні апаратні та програмні вимоги: Система повинна функціонувати відповідно до визначених мінімальних апаратних вимог для Android-пристроїв (Snapdragon 400, 2 ГБ ОЗУ) та ПК (Pentium N5000, 4 ГБ ОЗУ), а також програмних вимог (Android 10.0+, сучасні веб-браузери).
* Локалізація інтерфейсу: Інтерфейс системи повинен підтримувати локалізацію (українська та англійська мови) з можливістю розширення, як це вже зазначено в нефункціональних вимогах.

## 3.7 Other Requirements

* Система повинна підтримувати одночасну роботу не менше 1000 активних користувачів без помітного зниження продуктивності.
* Система повинна відповідати законодавству про захист авторських прав у країнах, у яких вона функціонує, та підлаштовуватися під любі зміни в цих законодавств.

# 

# 4. Analysis Models

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

## 4.1 Use Case Diagrams

### 4.1.1 Use Case android частини системи

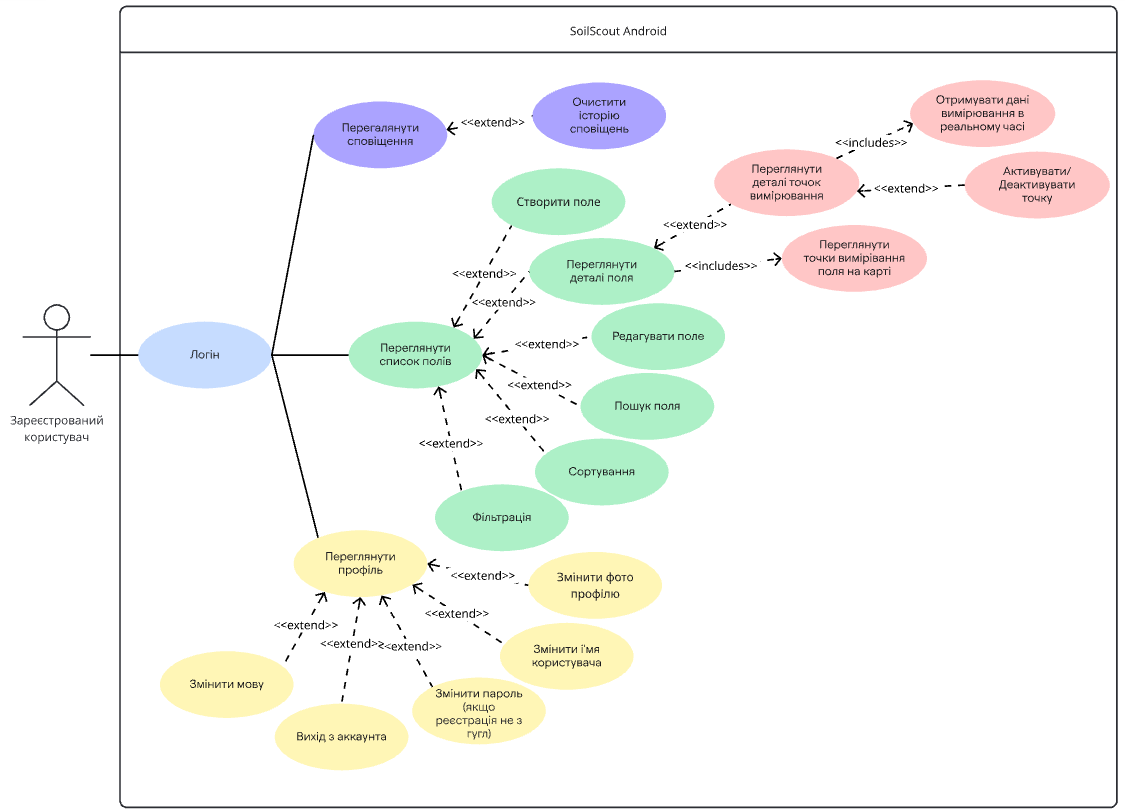
****

Рисунок 4.1 - Use case діаграма android частини системи

### 4.1.2 Use Case web частини системи

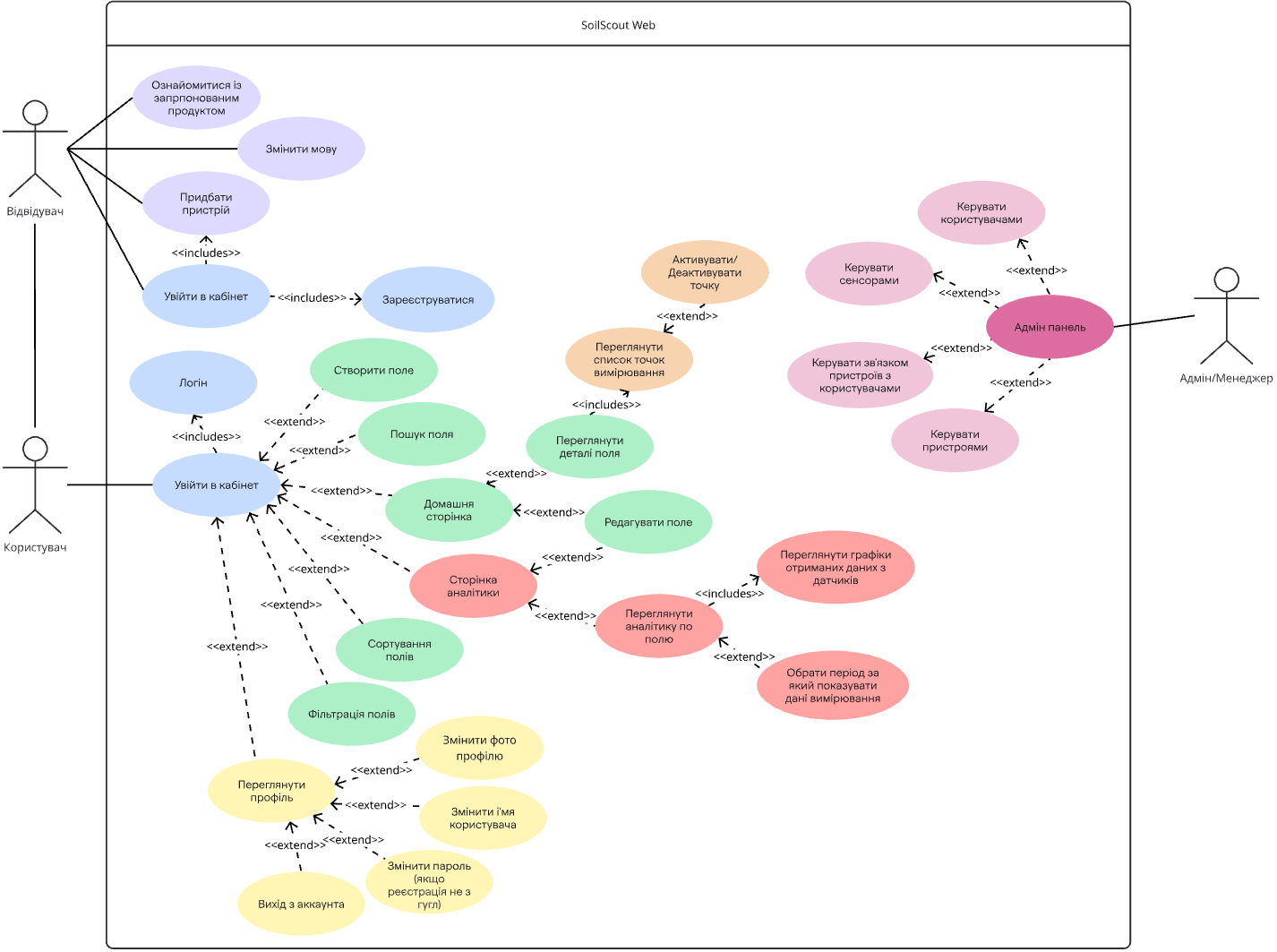


Рисунок 4.2 - Use case діаграма web частини системи

## 4.2 Database Design

### 4.2.1 Logical Database Schema

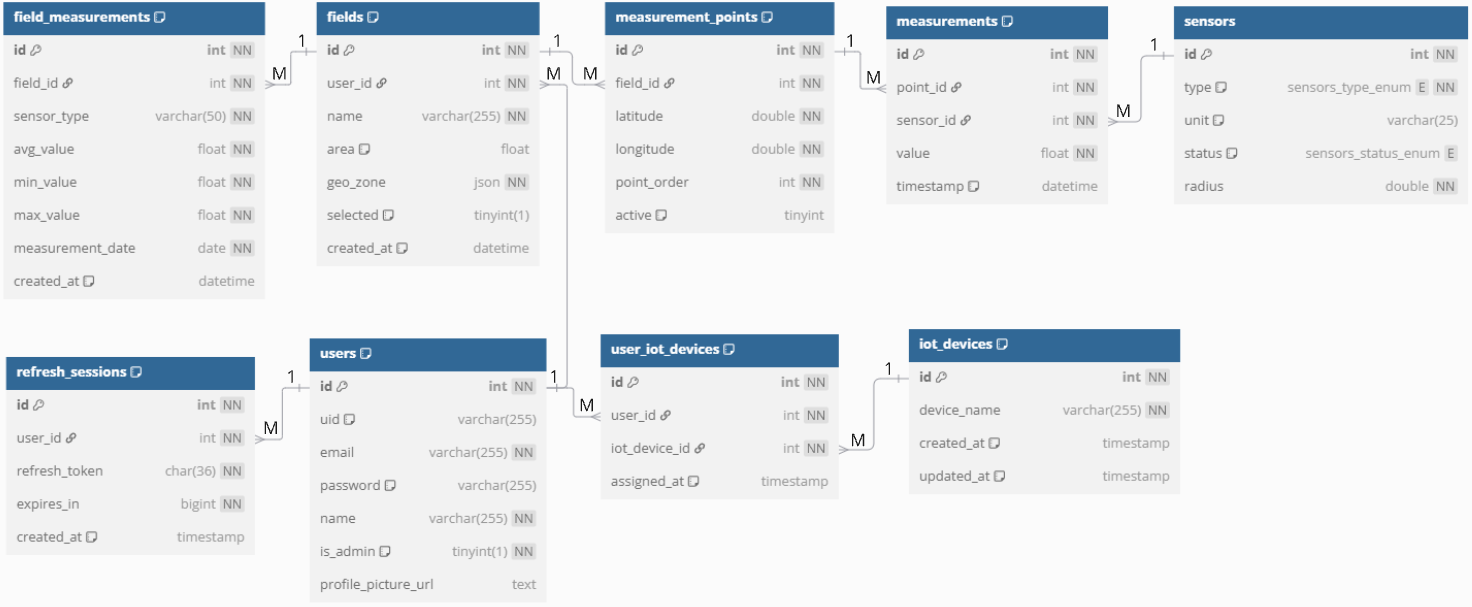


Рисунок 4.3 – Логічна схема бази даних

## 4.3 Sequence Diagrams

## 4.4 State-Transition Diagrams (STD)

## 4.5 Data Flow Diagrams (DFD)

# 5. Change Management Process

*Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.*

**A. Appendices**

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

**A.1 Appendix 1**

**A.2 Appendix 2**