

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри

Інформаційних систем і технологій

(назва кафедри)

Швиденко М.З.

(підпис)

“17” травня 2025 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему

Система моніторингу стану рослин

Спеціальність 126 “Інформаційні системи та технології”

Гарант освітньої програми

к.е.н., доцент

(науковий ступінь та вчене звання)

Мокрієв Максим Володимирович

(підпис)

(ПІБ)

Керівник кваліфікаційної роботи

Професор, д.т.н.

(науковий ступінь та вчене звання)

Смолій Вікторія Миколаївна

(підпис)

Виконав

Рой Микола Станіславович

(підпис)

Київ 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Завідувач кафедри інформаційних
систем і технологій

Швиденко М.З

Підпис ініціали та прізвище _____ 2025 р.

ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи
студенту(ці) Рою Миколі Станіславовичу

Спеціальності 126 “Інформаційні системи та технології”

1. Тема роботи: **“Система моніторингу стану рослин”**

Затверджена наказом ректора від 16.12.2024 р. № 2245-С

2. Термін подання завершеної роботи на кафедру - (01.06.2025)
3. Вихідні данні: значення температури повітря та ґрунту, рівень вологості повітря і ґрунту, зафіксовані у форматі JSON, а також візуальні зображення з камери. Усі дані можуть бути представлені користувачу у графічному інтерфейсі у вигляді таблиць або діаграм для подальшого аналізу.
4. Перелік питань, що розглядаються:
1. Аналіз проблеми моніторингу стану рослин у домашніх умовах
 2. Формування технічного завдання
 3. Вибір апаратних засобів
 4. Обґрунтування архітектури системи
 5. Календарний план

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми моніторингу стану рослин у домашніх умовах	20 лютого 2025	Успішно
2	Формування технічного завдання	15 січня 2025	Успішно
3	Обґрунтування архітектури системи	14 квітня 2025	Успішно
4	Обґрунтування архітектури системи	10 березня 2025	Успішно

Керівник кваліфікаційної роботи _____ / професор, д.т.н. В.М.Смолій
підпис ПБ, вчене звання та ступінь

Завдання прийняв до виконання _____ / Рой М.С.

підпис

ПБ

Дата отримання завдання 16.12.2024

РЕФЕРАТ

Тема бакалаврської кваліфікаційної роботи: “ Системи моніторингу стану рослин”.

Автор роботи: Рой Микола Станіславович.

Керівник роботи: Смолій Вікторія Миколаївна.

Пояснювальна записка: 58 с., 11 рис., 2 дод., 20 джерел.

Графічна частина: 13 презентаційних слайдів.

Бакалаврська кваліфікаційна робота присвячена розробці інформаційної системи моніторингу стану кімнатних рослин. Основною метою є створення програмного рішення, яке дозволяє в реальному часі здійснювати спостереження за рослиною, зчитувати екологічні параметри середовища та зберігати результати у зручному форматі для подальшого аналізу.

Система базується на поєднанні апаратного та програмного забезпечення. Апаратна частина включає сенсори температури (DHT22, DS18B20), вологості повітря та ґрунту, а також камеру для відеоспостереження. Програмна реалізація побудована на мові Java із застосуванням таких технологій, як JavaFX [19] для створення інтерфейсу користувача, OpenCV для роботи з відеопотоком, jSerialComm для комунікації з мікроконтролером, а також Gson для збереження даних у форматі JSON.

Система кожні 6 годин автоматично фіксує отримані показники та зберігає їх на локальному пристрої. Користувач має змогу переглядати поточні значення, переглядати відео з камери та аналізувати історію змін стану рослини. Робота системи не вимагає постійного підключення до Інтернету, що робить її зручною для домашнього використання

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	6
ВСТУП	7
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ 9	
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей.....	9
1.2 Аналіз наявних інформаційних технологій предметної області.....	12
1.3 Визначення вимог до інструментальних засобів та розробка технічного завдання	20
1.4 Висновок до першого розділу.....	23
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	25
2.1 Моделювання предметної області.....	25
2.2 Архітектура програмного забезпечення	29
2.3 Опис функціональних модулів: аналітика, рекомендації, база знань, користувачі.....	31
2.4 Вибір засобів розробки: мова програмування, фреймворки, СУБД	34
2.5 Проєктування бази даних: ER-діаграма, структура таблиць.....	36
2.6 Опис інтерфейсу користувача.....	39
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	41
3.1 Реалізація програмних модулів та інтеграція з базою даних	41
3.2 Розробка алгоритмів обробки даних та перехід на прямий ефір для спостереженням.....	45
3.3 Проведення тестування системи	47
3.4 Впровадження системи на тестовому квітнику	49

3.5	Оцінка результатів та аналіз ефективності.....	50
3.6	Висновки за результатами впровадження	53
	ВИСНОВКИ.....	54
	СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
	Додаток А.....	58
	Додаток Б	60

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- API — інтерфейс прикладного програмування (Application Programming Interface)
- JSON — текстовий формат обміну структурованими даними (JavaScript Object Notation)
- UI — інтерфейс користувача (User Interface)
- UX — досвід користувача (User Experience)
- NDVI — нормалізований вегетаційний індекс (Normalized Difference Vegetation Index)
- BSI — індекс яскравості ґрунту (Bare Soil Index)
- JVM — віртуальна машина Java (Java Virtual Machine)
- Swing — набір компонентів графічного інтерфейсу користувача в Java
- Rule — правило з бази знань для формування агрономічної поради
- Field — сутність, що описує агрополе користувача
- Recommendation — згенерована система поради на основі даних
- Weather — погодні параметри, отримані через зовнішній API
- Entropy — міра структурної складності зображення (ентропія Шеннона)
- DryIndex — індекс сухості зображення на основі аналізу пікселів
- HTML Report — звіт у форматі HTML, що формується системою
- Deployment Diagram — UML-діаграма розгортання програмного забезпечення
- ER-діаграма — діаграма зв'язків сутностей (Entity-Relationship Diagram)
- IDEF0 — метод функціонального моделювання для опису процесів системи
- IDEF5 — метод побудови онтологій у складних інформаційних системах
- BPMN — нотація моделювання бізнес-процесів (Business Process Model and Notation)

ВСТУП

У сучасному світі дедалі більше людей цікавляться вирощуванням рослин у домашніх умовах — як з естетичних міркувань, так і з прагнення покращити мікроклімат у житлі. Однак догляд за кімнатними рослинами вимагає уваги, досвіду й часу, що не завжди доступно в умовах напруженого ритму життя. Неправильний полив, недостатнє освітлення чи несприятливі температурні умови можуть призвести до захворювання або загибелі рослин.

Актуальним рішенням є створення інтелектуальної системи моніторингу, яка дозволяє автоматично контролювати стан кімнатних рослин і повідомляти власника про необхідність дій: поливу, провітрювання чи зміни місця розташування. Така система дає змогу значно полегшити догляд за рослинами та підвищити ефективність вирощування.

У межах даної роботи розроблено програмно-апаратний комплекс, який дозволяє збирати дані про температуру повітря, температуру ґрунту, вологість повітря та ґрунту за допомогою відповідних сенсорів. Для додаткового спостереження за зовнішнім виглядом рослини інтегровано камеру реального часу. Зібрані дані автоматично зберігаються кожні 6 годин, а на їх основі система генерує прості рекомендації щодо догляду.

Метою роботи є створення зручного та доступного інструменту для власників кімнатних рослин, який забезпечує постійний контроль стану зелених насаджень без потреби в спеціальних знаннях. Система є автономною, не потребує доступу до Інтернету та зберігає історію вимірювань у форматі JSON.

Об'єктом дослідження є процес догляду за кімнатними рослинами в умовах обмеженої присутності людини.

Предметом дослідження — інформаційна система збору, обробки та аналізу даних від сенсорів і камери для домашнього використання.

Розроблена система може стати корисною не лише для квітників-аматорів, а й для офісів, навчальних закладів або оранжерей, де важливо підтримувати оптимальні умови для зелених насаджень.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Україні дуже багато людей тримають рослин але не всі мають бажання за ним доглядати тим більше після робочого дня. Моя Система організована на принципах приватної власності, самозайнятості та локалізованого управління. Типова структура системи моніторингу стану рослин охоплює адміністративну функцію, , забезпечення даними, облік та контроль, а також інформаційно-консультаційний супровід . У реальній практиці така структура часто є неформалізованою, що ускладнює ефективне прийняття управлінських рішень, оперативне реагування на зміну умов виробництва та впровадження інноваційних агротехнологій .

У сучасних міських умовах дедалі більше людей тримають кімнатні та декоративні рослини як елемент озеленення інтер'єру, з метою покращення мікроклімату, зниження рівня стресу або просто як хобі. Проте догляд за рослинами потребує регулярного контролю температури, рівня вологості ґрунту та повітря, а також вчасного поливу, що є складним для більшості людей через нестачу часу, знань або досвіду.

Система моніторингу стану рослин у домашньому середовищі має на меті автоматизувати рутинні процеси нагляду за рослинами, зробити догляд більш об'єктивним, точним та менш залежним від інтуїції власника. Така система повинна отримувати дані від сенсорів температури та вологості повітря й ґрунту, обробляти ці дані та формувати прості, зрозумілі рекомендації. Крім того, додатковим засобом спостереження виступає камера, яка дозволяє візуально оцінювати стан рослини, наприклад, наявність в'янення, зміни кольору або ознак ураження.

На відміну від систем, орієнтованих на агропромисловий сектор, домашнє застосування висуває інші вимоги: простота у використанні, доступність, автономність, компактність та можливість інтеграції в інтер'єр. Крім того,

важливими є інтуїтивно зрозумілий інтерфейс та мінімальна кількість взаємодій з боку користувача.

Типова структура такої системи складається з таких компонентів:

- модуль зчитування сенсорних даних (температура, вологість);
- модуль візуального спостереження через камеру;
- програмна логіка генерації рекомендацій;
- система періодичного збереження даних (наприклад, кожні 6 годин);
- база знань з правилами догляду;
- модуль перегляду історії показників.

У структурному плані система поєднує апаратні засоби (сенсори, камера) та програмне забезпечення (Java-додаток із базою знань у форматі JSON). У функціональному плані вона повинна забезпечити стабільну роботу, надійне збереження даних, зручний доступ до інформації та просте розширення (наприклад, додавання нових типів рослин або порад).

Таким чином, актуальність створення системи моніторингу стану кімнатних рослин полягає в потребі допомогти користувачам забезпечити належні умови догляду за зеленими насадженнями без спеціальних знань і складного обладнання.

На рис. 1.1 представлено узагальнену функціональну схему взаємодії інформаційних компонентів системи моніторингу стану рослин системи.

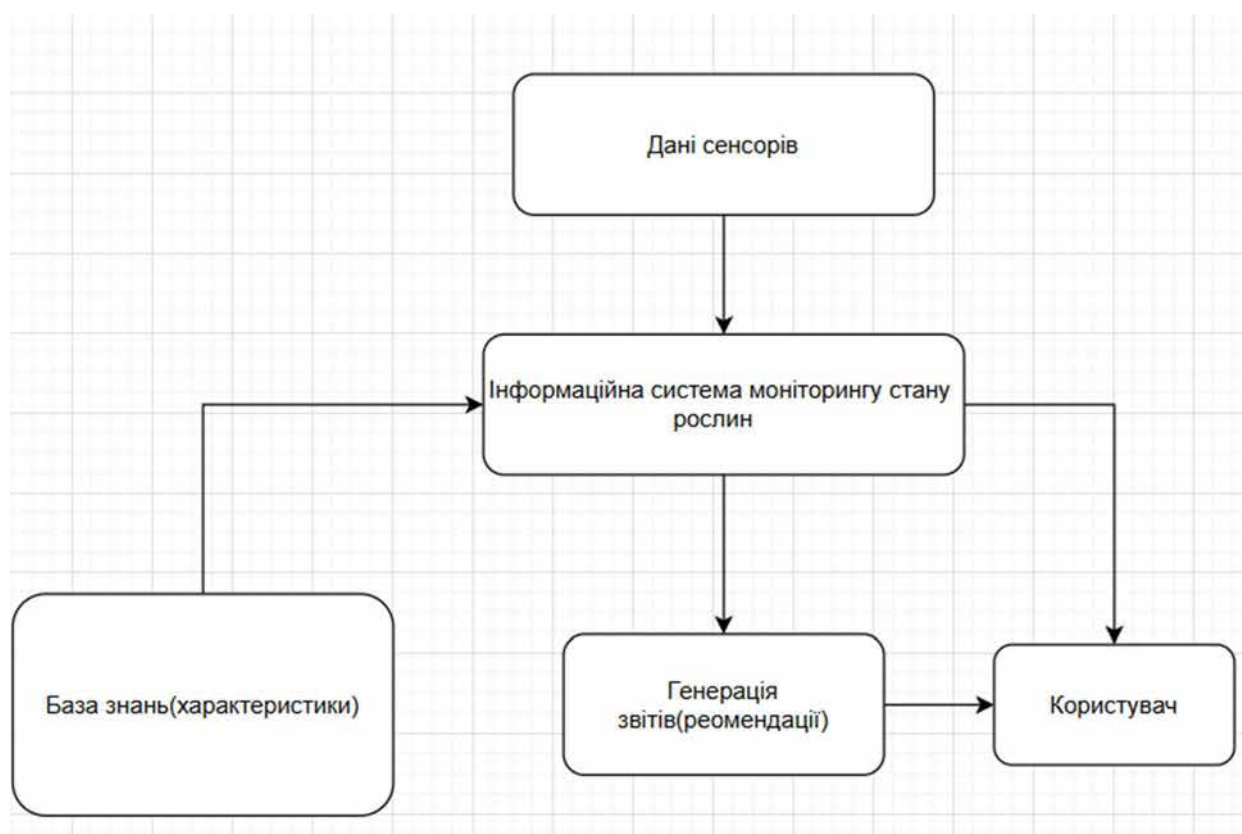


Рисунок 1.1 – Структура схеми взаємодії інформаційних компонентів

Вхідним елементом виступають агрономічні дані, що надходять до інформаційної системи з датчиків та візуальні з камери. Центральний елемент — системи моніторингу стану рослин інформаційна система — виконує роль посередника між даними та кінцевим користувачем. На основі внутрішньої експертної бази система формує персоналізований звіт. Вихідний елемент — Власник рослини — отримує структуровану та практично застосовну інформацію.

Водночас існує низка системних проблем, які стримують ефективно впровадження дорадчих рішень.

Варто відзначити, що предметна область системи моніторингу стану рослин характеризується складною структурою знань, багатоканальними потоками даних та високими вимогами до персоналізації рекомендацій. Це формує потребу у створенні автоматизованої системи, здатної не лише акумулювати знання, а й адаптивно їх застосовувати на практиці в режимі реального часу.

1.2 Аналіз наявних інформаційних технологій предметної області

Під час розробки власної системи моніторингу стану рослин я провів аналіз існуючих інформаційних технологій, які вже застосовуються у цій сфері. Метою цього аналізу було вивчити актуальні рішення, виявити їхні переваги та обмеження, а також зрозуміти, які підходи можна адаптувати або удосконалити у власній реалізації.

Нажаль провівши аналіз я не знайшов нічого подібного на мою тему, то му що найбільш популярним фактором є моніторинг стану полів тому що фермерам та власникам полів складно контролювати вологість ґрунту та стан посівів на двох та більше полів. Тому як приклад як приклад оформлення та ознайомлення я вибрав цей альтернативний варіант.

Я ознайомився з низкою комерційних платформ, серед яких — Storjo, FieldView, EOS Crop Monitoring та OneSoil. Ці сервіси є багатофункціональними рішеннями для великих аграрних підприємств і пропонують глибоку аналітику на основі супутникових даних, метеомоделей, IoT-датчиків та історичних записів про поля. Вони дозволяють відслідковувати стан рослин у реальному часі, отримувати індекси вегетації [2-6], аналізувати ризики та будувати довгострокові прогнози. Однак у ході аналізу я звернув увагу на те, що більшість з них вимагають стабільного підключення до Інтернету, наявності серверної інфраструктури, досвіду інтеграції та часто передбачають підписку або ліцензію, що суттєво обмежує їхню доступність для невеликих або індивідуальних користувачів.

Крім промислових рішень, я дослідив також мобільні застосунки, орієнтовані на приватних користувачів, садівників та власників присадибних ділянок. Додатки на кшталт Plantix, Blossom або Planta надають базові поради щодо поливу, підживлення або лікування рослин, а також можуть аналізувати фото рослини для виявлення можливих проблем. Хоча такі сервіси мають привабливий інтерфейс і зручні в користуванні, я помітив, що вони зазвичай не враховують локальних кліматичних особливостей, мають обмежені можливості

налаштування, не дозволяють працювати без Інтернету і часто не адаптовані до україномовного користувача.

Особливу увагу я також звернув на наукові розробки у сфері агроінформатики. У багатьох дослідженнях пропонуються системи, які базуються на експертних правилах, моделях штучного інтелекту або статистичному аналізі. Проте більшість з них залишаються у статусі експериментальних прототипів, мають складну реалізацію або не мають користувацького інтерфейсу. Я зробив висновок, що подібні проєкти хоча й цікаві з наукової точки зору, однак на практиці часто є непридатними для безпосереднього використання у господарствах без додаткового доопрацювання.

Після цього аналізу я дійшов висновку, що існує певна "прогалина" між складними, дорогими рішеннями для агрохолдингів і простими мобільними застосунками для домашнього використання. Немає достатньої кількості локалізованих, автономних, простих у використанні систем, які б дозволяли аграріям чи дорадникам ефективно моніторити стан рослин без зайвих технічних складнощів.

З урахуванням цього я вирішив створити власну систему моніторингу, яка працює без необхідності постійного підключення до Інтернету, зберігає дані локально у форматі JSON, і водночас дозволяє гнучко редагувати базу знань через звичайні текстові файли. Для обробки даних я реалізував знанняорієнтовану модель на основі правил, що дало змогу уникнути складних обчислень або потреби у машинному навчанні на початковому етапі. У перспективі така архітектура дозволяє масштабувати рішення, інтегрувати зовнішні джерела метеоданих, додати веб-інтерфейс чи мобільну версію.

Кільцеві зони навколо сенсорів демонструють градацію загрози (зелена — нормальні умови, червона — критичні показники), що дозволяє оперативно ідентифікувати ділянки з високим ризиком розвитку хвороб.

FieldView — це програмне забезпечення, орієнтоване на інтеграцію з аграрною технікою та детальну картографію врожайності. Рис. 1.2 ілюструє

можливості системи щодо формування диференційованих карт на основі NDVI-індексу.

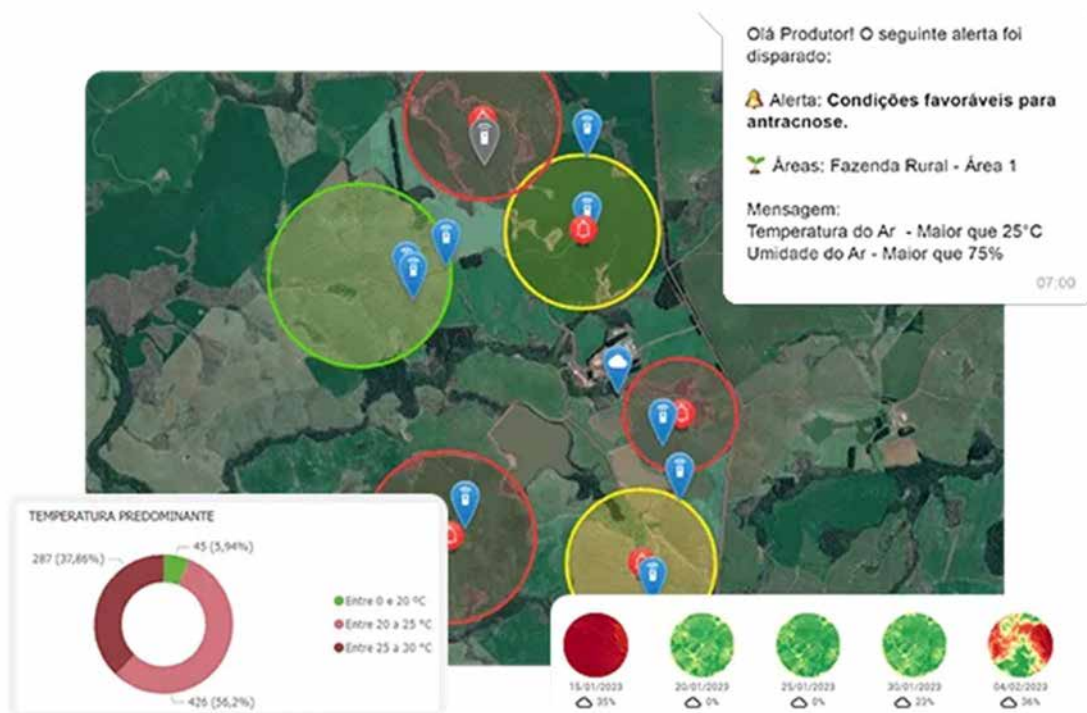


Рисунок 1.2 - Картографія зон врожайності у FieldView

Інтерфейс мобільного додатку забезпечує доступ до історії польових операцій, поточного стану посівів та погодних умов у реальному часі, що підвищує гнучкість планування агрономічних заходів .

EOS Crop Monitoring — це одна з провідних цифрових платформ для супутникового моніторингу сільськогосподарських угідь. Я детально ознайомився з її можливостями під час аналізу існуючих інформаційних технологій у сфері моніторингу стану рослин, щоб порівняти її функціональність із завданнями мого проєкту.

EOS Crop Monitoring надає користувачам доступ до високоточних супутникових знімків та аграрної аналітики в режимі майже реального часу. Платформа дозволяє відслідковувати стан вегетації сільськогосподарських культур за допомогою індексів NDVI, NDWI, MSAVI, реєструвати зміни поля, виявляти аномалії та оцінювати ефективність обробітку. Однією з найважливіших функцій, яку я відзначив, є можливість створення зон

агровтручання та диференційованого внесення добрив. Користувачі також мають доступ до погодних даних, історії посівів, прогнозів та облікових звітів.

EOS Crop Monitoring орієнтована переважно на агрокомпанії, агрономів-консультантів, трейдерів та агростраховиків. Платформа працює через вебінтерфейс і підтримує інтеграцію з іншими системами через API. Інтерфейс має багатомовну підтримку, включаючи українську, що робить її доступною для вітчизняного користувача.

Водночас, при всій своїй функціональності, EOS Crop Monitoring має ряд обмежень, які я врахував у своїй роботі. По-перше, вона є хмарним сервісом і вимагає постійного доступу до інтернету. По-друге, її інтеграція передбачає наявність електронних карт полів у відповідному форматі, що може бути незручно для дрібних фермерів. По-третє, хоча базові функції доступні безкоштовно, розширений функціонал потребує комерційної підписки.

У процесі проектування власної системи я взяв до уваги ідею надання агровиробнику доступу до релевантної інформації у простій формі, однак обрав інший підхід: створення автономної, настільної системи, яка працює без зовнішніх сервісів і зберігає дані локально (рис. 1.3). Таким чином, моя система позиціонується як легка альтернатива великим платформам на кшталт EOS, орієнтована на освітні, консультативні або фермерські застосування в умовах обмежених ресурсів.

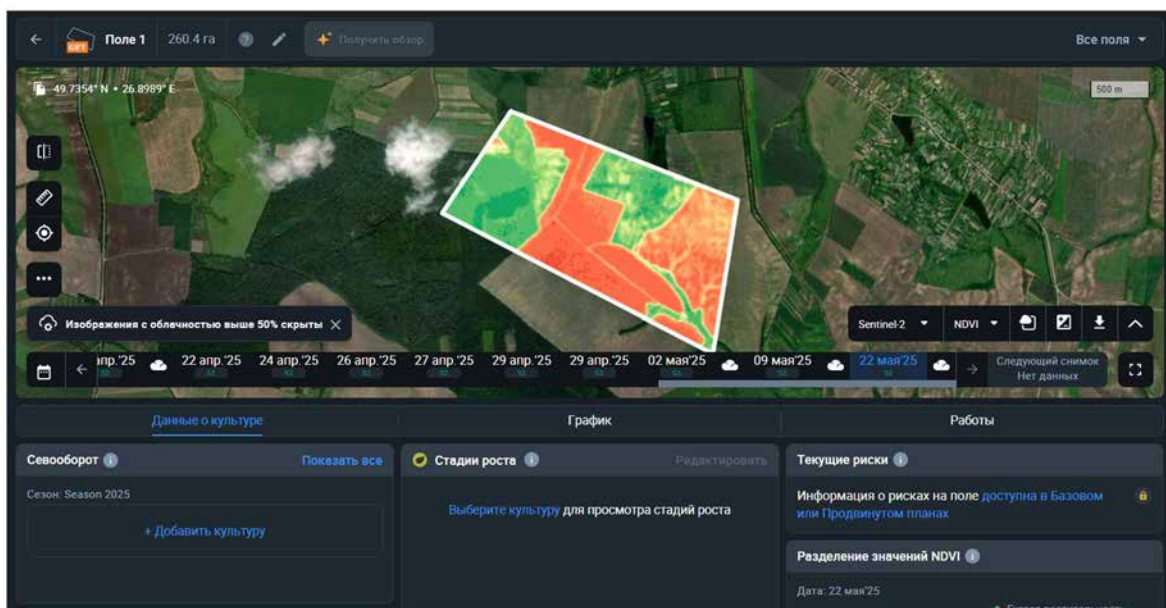


Рисунок 1.3 - Інтерфейс управління полями у платформі EOS Crop Monitoring

OneSoil — це безкоштовна платформа для точного землеробства, яка надає інструменти для супутникового моніторингу стану полів і управління аграрними ділянками. У межах мого аналізу інформаційних технологій у сфері моніторингу стану рослин я детально вивчив функціональність OneSoil, щоб оцінити її практичну цінність та співвідношення з концепцією моєї власної системи.

Однією з головних особливостей OneSoil є автоматичне розпізнавання меж полів за допомогою супутникових знімків. Користувачеві не потрібно вручну креслити контури ділянок — система самостійно ідентифікує їх і додає до персонального кабінету. Далі користувач має змогу відстежувати індекси вегетації [1] (зокрема NDVI), оцінювати стан рослинності та переглядати історію змін на полі протягом сезону. Крім того, платформа пропонує інструменти для створення зон диференційованого внесення добрив.

Сервіс працює через вебінтерфейс і мобільний додаток. Він не потребує оплати, що робить його привабливим для малих і середніх фермерських господарств. Також важливо, що OneSoil пропонує багатомовну підтримку, у тому числі українською, що робить систему доступною для локального користувача без бар'єру в мові.

Втім, під час аналізу я виявив і низку обмежень. Платформа працює виключно через інтернет, а отже, не може бути використана в автономному режимі. Також її основна увага зосереджена на великих сільськогосподарських полях та індексах знімків, але не передбачає гнучкого редагування логіки формування рекомендацій або створення власних правил. Вона не призначена для роботи з невеликими квітковими або садовими ділянками, які не завжди потрапляють у фокус супутникових даних. Крім того, OneSoil не підтримує локальне збереження історії або незалежну роботу на комп'ютері без доступу до серверу.

У порівнянні з такими хмарними платформами, як OneSoil, моя система моніторингу стану рослин має іншу філософію. Вона призначена для автономного використання без підключення до інтернету, з локальним збереженням інформації у форматі JSON, власною базою знань, яку можна редагувати вручну, та можливістю швидкої адаптації під будь-який тип господарства. Це робить її більш придатною для роботи в польових умовах, у навчальних закладах або як локальний інструмент для аграрного консультування без складної інфраструктури на рис. 1.4.

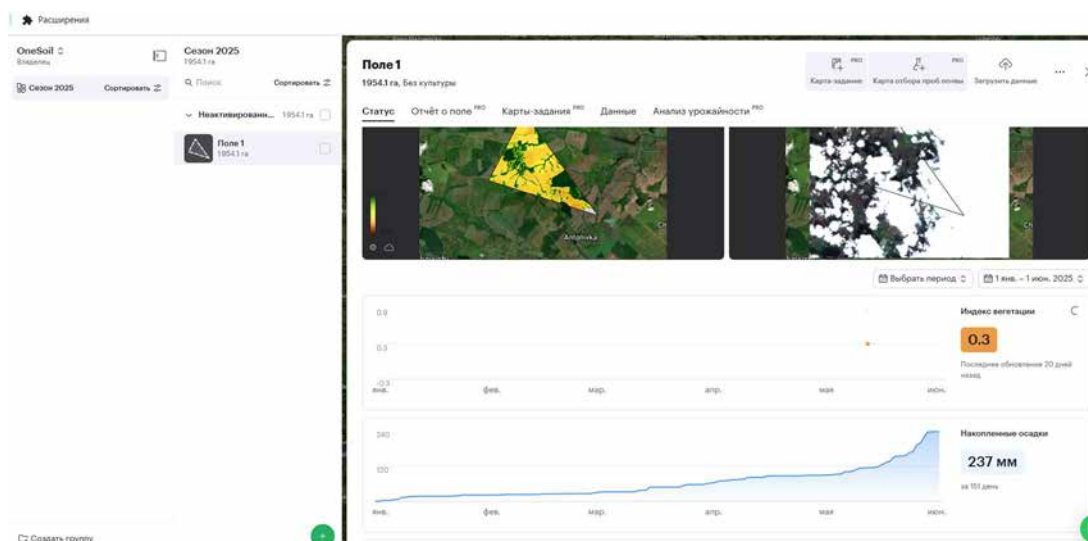


Рисунок 1.4 - Зони продуктивності поля у OneSoil

Платформа дозволяє створювати диференційовані карти висіву та добрив, інтегрується з GPS-обладнанням та забезпечує візуальний контроль за виконанням технологічних карт.

Для виявлення переваг і обмежень наявних програмних рішень проведено порівняння чотирьох систем: AgroSmart, Agrivi, FieldView та Cropwise Operations. Основними критеріями оцінювання обрано функціональність, зручність інтерфейсу (UI/UX), наявність україномовної локалізації та доступність для малих системи моніторингу стану рослин. Дані узагальнено в табл. 1.1.

Таблиця 1.1 – Порівняння існуючих аграрних IT-рішень за ключовими критеріями

Критерій	Cropio	FieldView	EOS Crop Monitoring	OneSoil
Тип платформи	Комерційна, корпоративна	Комерційна, для агрохолдингів	Freemium / SaaS	Безкоштовна, масова
Основний функціонал	Супутниковий моніторинг, техніка, логістика	Супутникові карти, сівозміни, техніка	Індекси вегетації, прогноз, аналітика	NDVI, розпізнавання полів, простий моніторинг
Індекси вегетації (NDVI, EVI)	так	так	так	так
Робота без інтернету	ні	ні	ні	ні
Локальне збереження даних	ні	ні	ні	ні
Мобільна версія	так (обмежено)	так	так	так

Продовження таблиці 1.1

Можливість експорту звітів	так	так	так (PDF, CSV)	ні(лише перегляд)
Метеодані / прогноз погоди	так	так	так	так
Мова інтерфейсу (включаючи українську)	ні	ні	так	так
Редагування логіки/правил користувачем	ні	ні	ні	ні
Вартість / доступність	Ліцензія / підписка	Платна	Частково безкоштовна, розширена — платна	Повністю безкоштовна
Призначення	Великі агропідприємства	Великі господарства, корпоративний рівень	Професійне консультування, агробізнес	Малі фермери, освітнє використання

Аналіз свідчить, що більшість платформ демонструють високий рівень функціональності, однак потребують значних фінансових ресурсів для повноцінного використання або орієнтовані на великотоварне виробництво .

Водночас інтерфейси користувача часто перевантажені або не локалізовані, що знижує ефективність взаємодії для україномовного фермерського сегменту.

Висновки про недоліки та невирішені завдання

Незважаючи на широке впровадження аграрних ІТ-рішень, існує низка системних обмежень, які перешкоджають їхньому ефективному застосуванню у дрібному фермерському виробництві:

1. відсутність локалізації українською мовою у більшості систем, що ускладнює розуміння та використання для фермерів без англомовної підготовки.
2. Недоступність для малого бізнесу через високу ціну ліцензії, підписки або необхідність у спеціалізованому обладнанні .
3. Обмежена персоналізація порад, коли системи не враховують специфіку мікроклімату, ґрунтів або виробничих потужностей конкретного господарства .
4. Закритість архітектури та неможливість адаптації системи до локальних умов без втручання розробника .

Актуальним залишається завдання розробки інформаційної системи системи моніторингу стану рослин, що поєднує експертну підтримку, гнучкість у налаштуванні, україномовний інтерфейс та адаптацію до умов малого фермерства. Це дозволить зменшити цифровий розрив між високотехнологічними агровиробниками та дрібними господарствами, підвищивши ефективність агровиробництва в Україні загалом.

1.3 Визначення вимог до інструментальних засобів та розробка технічного завдання

Проектування інформаційної системи системи моніторингу стану рослин підтримки системи моніторингу стану рослин вимагає чіткого визначення її функціональних і нефункціональних вимог. Функціональні вимоги мають відображати очікувану поведінку системи з точки зору обробки, інтерпретації та представлення аграрної інформації. Основними компонентами, які повинна реалізувати система, є: агрегування даних з різних джерел (метеостанції,

спутникові знімки [4], польові сенсори), генерація рекомендацій на основі правил та моделей, формування короткострокових прогнозів, передача повідомлень користувачу та збереження історичних даних в структурованому форматі

Узагальнення функціональних вимог подано в табл. 1.2.

Таблиця 1.2 - Функціональні вимоги до системи моніторингу стану рослин

Компонент	Опис функціональності
Інформаційний блок	Отримання даних з API, пряме спостереження через камеру
Генерація порад	Автоматизоване формування рекомендацій з бази знань залежно від вхідних параметрів
Повідомлення	Сповіднення про ризики, рекомендації, зміни погодних умов через інтерфейс

Окрім функціональної складової, не менш важливою є реалізація нефункціональних вимог, які забезпечують стабільність, масштабованість, зручність у використанні та відповідність системи умовам реального агровиробництва. Високий рівень зручності інтерфейсу, підтримка різних пристроїв і браузерів, адаптивність до мобільних платформ, а також можливість подальшого розширення функціональності мають бути закладені на етапі архітектурного проектування

Узагальнені нефункціональні вимоги наведено в табл. 1.3

Таблиця 1.3 - Нефункціональні вимоги до системи

Вимога	Опис
Зручність	Інтуїтивний веб-інтерфейс, підтримка мобільних пристроїв
Масштабованість	Можливість додавання нових модулів (нові типи культур, джерела даних, поради)
Доступність	Робота в встановленій ПЗ
Надійність	Автоматичне збереження даних
Локалізація	Підтримка української мови

На основі функціональних та нефункціональних вимог було розроблено технічне завдання, яке чітко визначає структуру вхідних параметрів, формат вихідної інформації, обмеження реалізації, а також цільову аудиторію розробки. Технічне завдання розглядається як вихідна специфікація, що регламентує архітектурне проєктування, модульну декомпозицію та вибір технологічного стеку. Його реалізація передбачає підтримку як ручного, так і автоматизованого збору інформації, що надходить із зовнішніх джерел через API або формується у процесі користувацької взаємодії.

Для зручного візуального представлення структури технічного завдання було побудовано інтелект-карту (mind map), яка відображає основні логічні компоненти системи: блок вхідних даних, блок вихідної інформації, технічні обмеження та сегмент цільових користувачів. На рис. 1.3 продемонстровано взаємозв'язки між цими елементами, що дозволяє краще уявити узгодженість архітектурних вимог з функціональним призначенням майбутньої системи.

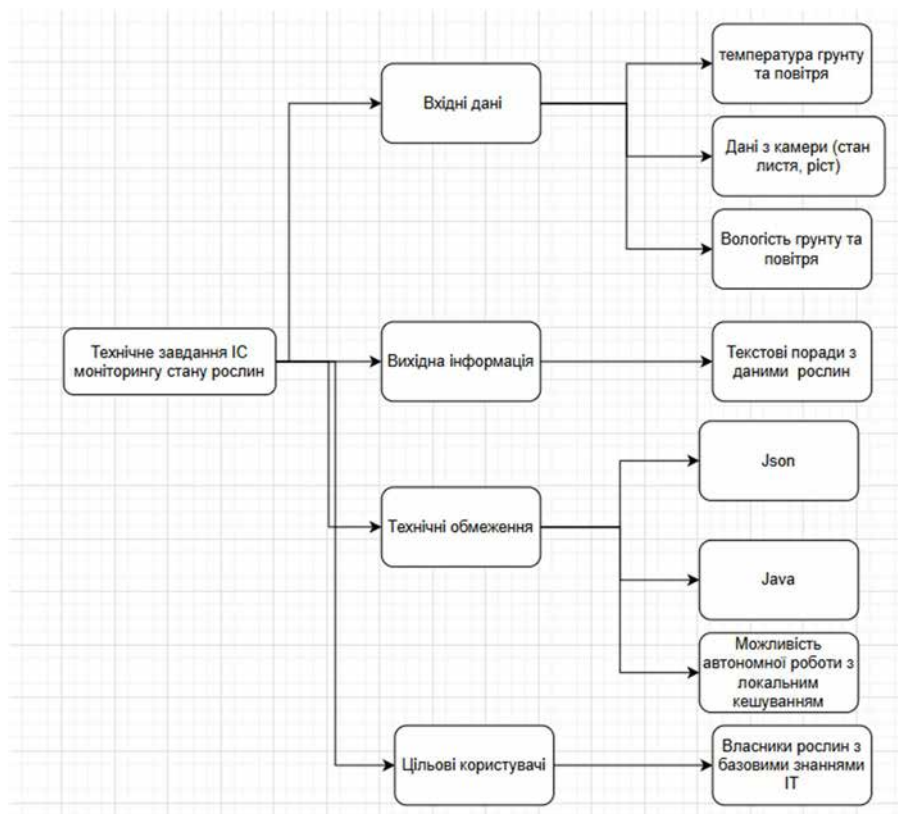


Рисунок 1.3 - Інтелект-карта технічного завдання системи моніторингу стану рослин

Розробка системи повинна враховувати ряд обмежень. Зокрема, реалізація має ґрунтуватись на технологіях із відкритим кодом, використовувати JSON для збереження в них даних користувача. Комунікація між модулями здійснюється через REST API. Клієнтська частина повинна працювати у веббраузері, бути сумісною з мобільними пристроями та не потребувати інсталяції додаткового ПЗ. Вибрані бібліотеки повинні забезпечувати обробку, агрегацію, візуалізацію та маршрутизацію даних у реальному часі.

Цільовими користувачами системи визначено власників персональних рослин в тобуті, які ще не мають досвід в догляді . Тому особлива увага приділяється зручності інтерфейсу, мінімізації кількості дій для отримання даних, а також автоматизації процесів, що не потребують втручання користувача.

1.4 Висновок до першого розділу

У результаті виконаного дослідження сформовано цілісне уявлення про предметну область системи моніторингу стану рослин в умовах цифрової трансформації агросектору. Проаналізовано структурні та функціональні особливості системи моніторингу стану рослин, зокрема їхню організаційну модель, інформаційні потоки та типові проблеми у доступі до актуальних агрономічних знань. Встановлено, що ключовими бар'єрами для впровадження цифрових рішень залишаються недостатня локалізація, висока вартість комерційних систем, а також відсутність персоналізації дорадчих послуг.

Оцінка сучасних ІТ-рішень (AgroSmart, FieldView, Agrivi, Cropwise) виявила високий рівень технічної реалізації, але водночас — обмежену доступність для малих фермерів та недостатню відповідність локальним умовам. Порівняльний аналіз за критеріями функціональності, зручності інтерфейсу, доступності та підтримки мови засвідчив наявність незадоволеного сегмента користувачів, потреби якого не покриваються існуючими комерційними продуктами.

На основі виявлених обмежень сформульовано функціональні та нефункціональні вимоги до майбутньої інформаційної системи системи моніторингу стану рослин. Основними завданнями такої системи визначено: агрегацію вхідних даних (поля, культури, погодні параметри), генерацію персоналізованих порад, візуалізацію погодних змін і технологічних рекомендацій, а також інтеграцію з доступними відкритими джерелами (API, сенсори, супутникові знімки [4,10]). Технічне завдання передбачає реалізацію веборієнтованої, адаптивної системи з відкритою архітектурою, що відповідає вимогам масштабованості, доступності та простоти використання для власників рослин із базовим рівнем цифрової грамотності .

Отримані результати лягають в основу архітектурного та функціонального проєктування, що буде реалізовано у наступному розділі, де буде побудовано структурну модель системи, визначено склад модулів та описано схеми взаємодії компонентів.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Моделювання предметної області

Моделювання предметної області є ключовим етапом на етапі передпроектного аналізу інформаційної системи, оскільки дозволяє формалізувати функціональні можливості, потоки взаємодії, поведінкові сценарії та зовнішні залежності системи. У контексті системи моніторингу стану рослин системи моніторингу стану рослин це моделювання забезпечує структуроване представлення ролей користувачів, їх цілей, логіки отримання рекомендацій та взаємозв'язків між внутрішніми модулями системи й зовнішніми джерелами даних. Для цього використано інструментарій мови UML — універсального засобу візуального моделювання, який дозволяє описати як статичну, так і динамічну складову предметної області.

На рис. 2.1 представлено діаграму прецедентів, яка ілюструє взаємодію зовнішніх акторів із системою.

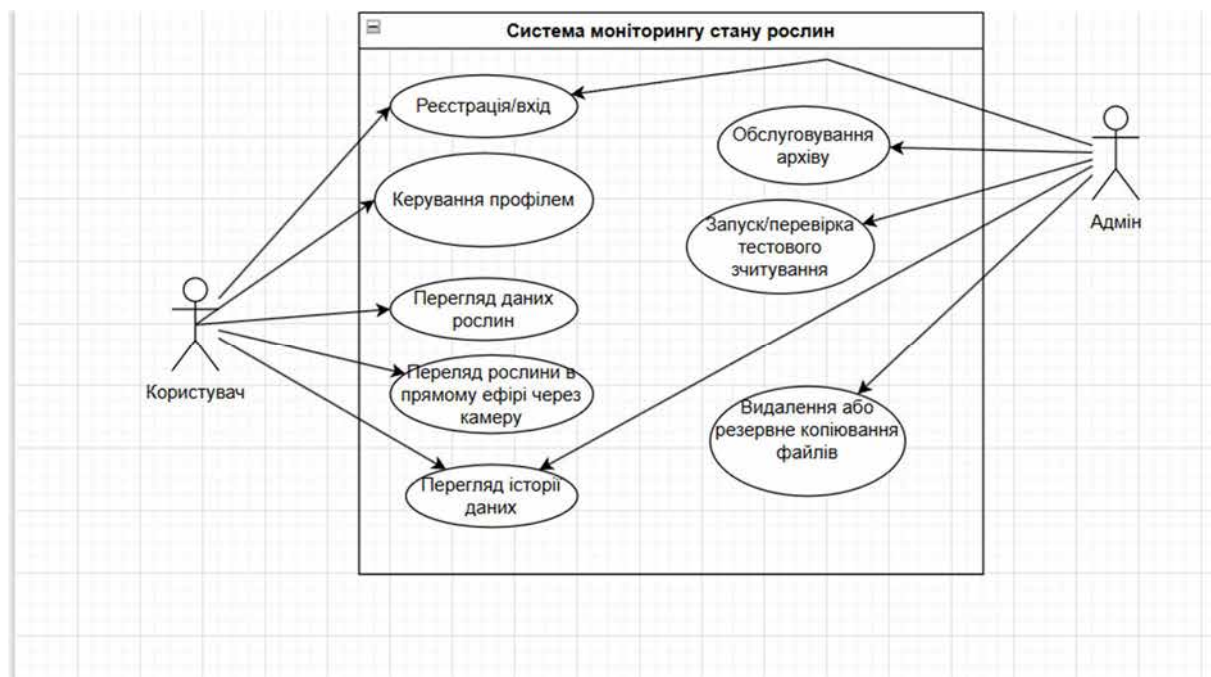


Рисунок 2.1 - Діаграма прецедентів системи моніторингу стану рослин

Користувачем системи моніторингу є власник кімнатної рослини, який взаємодіє з нею через зручний інтерфейс. Його головним завданням є спостереження за станом рослини без потреби постійного ручного контролю. Завдяки системі користувач може отримувати актуальну інформацію про

температуру повітря та ґрунту, рівень вологості навколишнього середовища й ґрунту, а також бачити зображення рослини в реальному часі за допомогою камери. Це дозволяє своєчасно виявляти зміни у зовнішньому вигляді рослини, наприклад, в'янення або пожовтіння листя.

Користувач має змогу переглядати історію зібраних даних, що допомагає відстежувати динаміку змін умов утримання. У системі також реалізована функція вибору типу рослини, що дозволяє налаштувати перегляд відповідно до конкретного виду. Водночас користувачу не потрібно самостійно налаштовувати обладнання чи структуру збереження даних — система працює автономно та з мінімальною потребою у втручанні. Це робить її доступною та зручною для використання навіть людьми без технічної підготовки.

У рамках динамічного аналізу предметної області побудовано діаграму послідовності (рис. 2.2), яка описує обмін повідомленнями між логічними компонентами системи в отримувannya звітів та перегляд з камери.

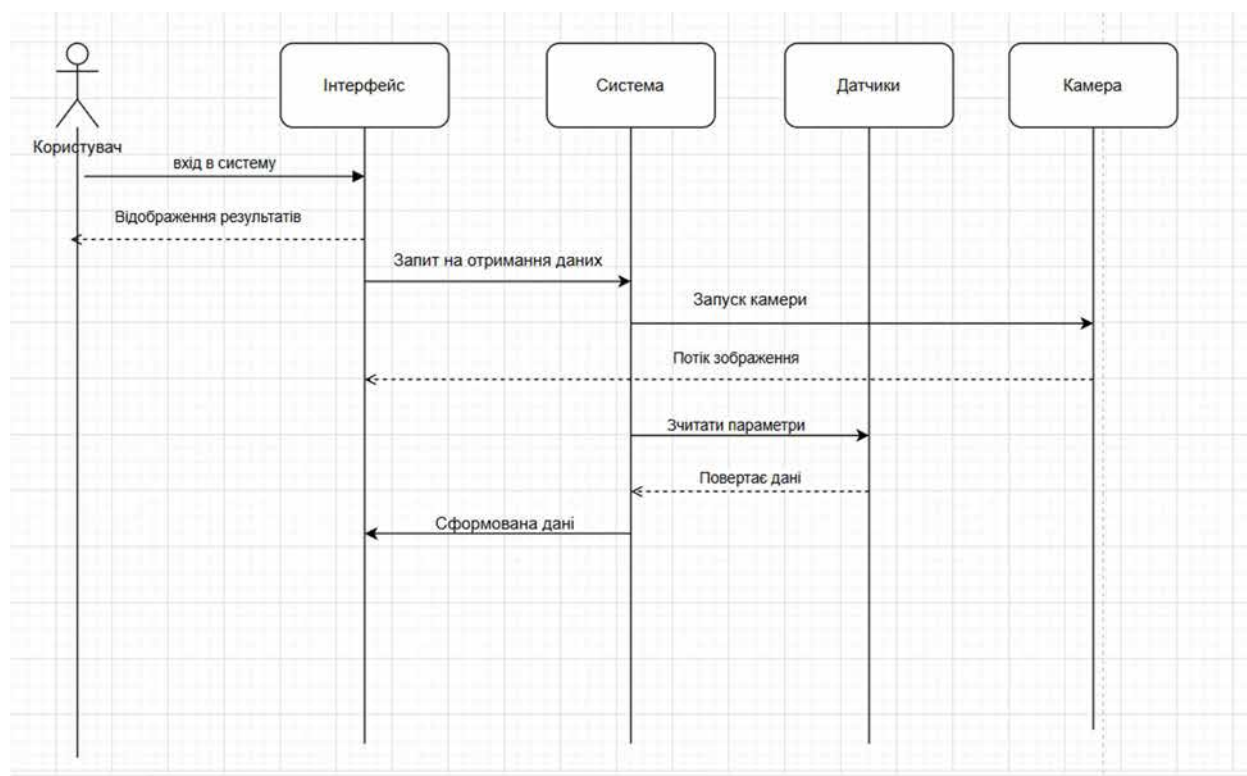


Рисунок 2.2 - Діаграма послідовності для отримувannya звітів та перегляд з камери

У процесі роботи системи моніторингу стану кімнатної рослини користувач взаємодіє з інтерфейсом, який у свою чергу ініціює взаємодію з іншими компонентами. Після запуску системи користувач бачить інтерфейс, що дозволяє переглянути актуальний стан рослини.

Інтерфейс надсилає запит до центральної частини системи, яка відповідає за логіку обробки. Система звертається до апаратних сенсорів: зчитує температуру повітря, температуру ґрунту, вологість повітря та вологість ґрунту. Отримані значення повертаються до інтерфейсу для відображення на екрані користувача.

Паралельно система активує модуль камери, яка транслює відео в реальному часі. Користувач може бачити поточний вигляд рослини, що дозволяє візуально контролювати її стан, не підходячи до неї.

У певні часові інтервали (наприклад, кожні 6 годин) система зберігає всі отримані сенсорні дані у JSON-файл. Це дозволяє формувати історію спостережень і, за потреби, аналізувати тенденції зміни параметрів середовища.

Таким чином, усі етапи — зчитування, виведення, відеоспостереження та збереження — виконуються автоматично без участі користувача, що робить систему зручною для повсякденного використання.

З метою опису логіки поведінки системи при виконанні бізнес-процесу запиту агрономічної поради побудовано діаграму активності (рис. 2.3).

Усе починається з ініціації запуску — користувач відкриває інтерфейс, після чого система автоматично намагається встановити з'єднання з фізичними сенсорами (температури та вологості повітря і ґрунту), а також камерою. Якщо з'єднання з усіма пристроями встановлено успішно, відбувається зчитування сенсорних даних.

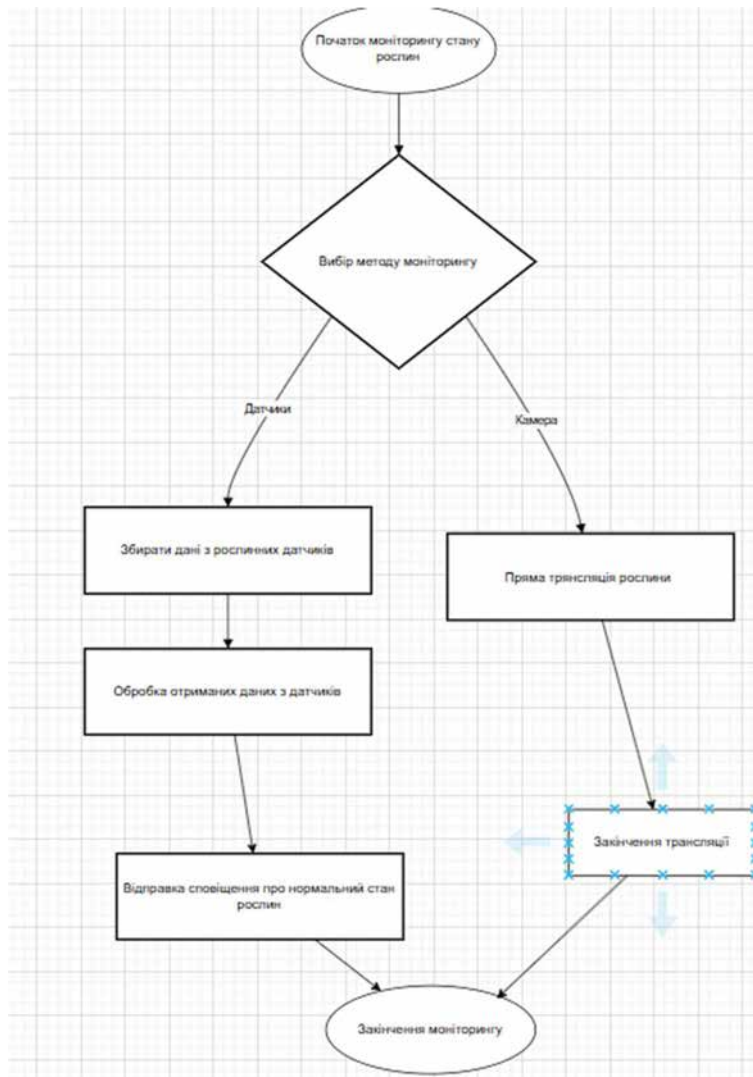


Рисунок 2.3 - Діаграма активності процесу моніторингу за рослиною

Після отримання значень від датчиків активується камера, яка транслює відео потік у реальному часі. Всі отримані дані — як числові, так і візуальні — відображаються в інтерфейсі користувача. Таким чином, користувач має змогу одночасно бачити актуальні показники середовища та візуальний стан рослини.

Далі система переходить у стан очікування, в якому вона постійно перевіряє, чи минув інтервал у 6 годин. Якщо так — відбувається автоматичне збереження даних у файл формату JSON, що містить часову мітку та всі актуальні сенсорні показники.

У разі, якщо підключення до сенсорів або камери з якихось причин не встановлюється, система повідомляє про помилку. При цьому користувач бачить відповідне повідомлення в інтерфейсі, а основний цикл зупиняється до усунення проблеми.

Завершенням процесу є або ручне закриття програми користувачем, або її подальше фонове функціонування з періодичним оновленням інформації.

2.2 Архітектура програмного забезпечення

Розробка програмного забезпечення системи моніторингу стану рослин базується на принципах модульності, розділення відповідальностей та масштабованості. В основу архітектурної моделі покладено класичний трирівневий підхід, який передбачає поділ системи на презентаційний рівень, рівень бізнес-логіки та рівень доступу до даних. Така структура забезпечує гнучкість розгортання, легкість у супроводі та можливість розширення функціональності в майбутньому без порушення цілісності системи.

На рисунку 2.4 представлено архітектуру програмного забезпечення, що демонструє взаємозв'язки між основними логічними блоками системи.

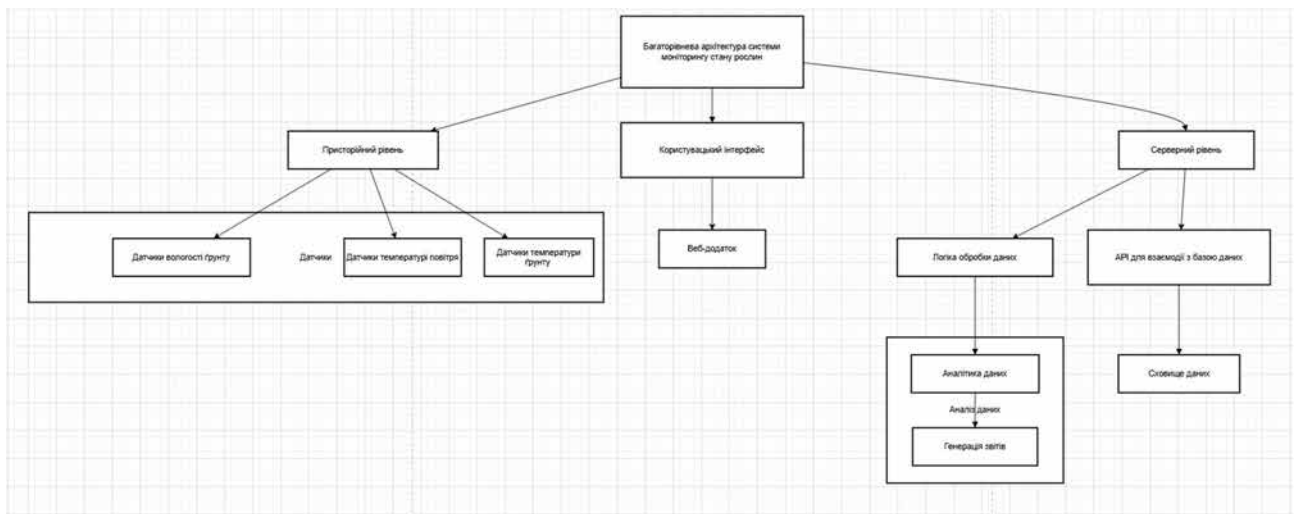


Рисунок 2.4 - Багаторівнева архітектура системи моніторингу стану рослин

У процесі проєктування системи моніторингу стану кімнатних рослин було вирішено використовувати багаторівневу архітектуру, яка забезпечує розділення логіки за функціональними шарами, що значно спрощує розробку, обслуговування та подальше масштабування системи. Такий підхід дозволяє чітко визначити відповідальність кожного модуля, забезпечити взаємозамінність компонентів, а також зменшити складність у разі модернізації системи.

Система складається з чотирьох логічно відокремлених рівнів:

Рівень 1. Фізичний рівень (рівень пристроїв)

Цей рівень складається з апаратного забезпечення, яке безпосередньо взаємодіє з навколишнім середовищем. До його складу входять:

- Сенсори:
 - DHT22 — вимірювання температури та вологості повітря;
 - DS18B20 — вимірювання температури ґрунту;
 - Capacitive Soil Moisture Sensor — вимірювання вологості ґрунту.
- Камера: USB-камера або Raspberry Pi Camera, що забезпечує візуальне спостереження за рослиною.
- Контролер: Arduino або Raspberry Pi, який зчитує дані з сенсорів і передає їх на персональний комп'ютер або інший обчислювальний пристрій через COM-порт або USB-з'єднання.

Цей рівень забезпечує первинний збір інформації про фізичний стан середовища, у якому знаходиться рослина.

Рівень 2. Рівень збору та передачі даних

Другий рівень відповідає за зв'язок між апаратною частиною та програмною логікою. Основне завдання цього рівня — забезпечити стабільне зчитування даних із сенсорів і камер та передати їх у центральний програмний модуль.

- Для зчитування даних із сенсорів використовується бібліотека `jSerialComm` (у середовищі Java), яка дозволяє працювати з COM-портами.
- Камера підключається до системи за допомогою `OpenCV`, що забезпечує обробку відеопотоку у реальному часі.

Цей рівень діє як міст між фізичними пристроями та логікою обробки, гарантуючи безперервне надходження даних.

Рівень 3. Рівень обробки та зберігання даних

Цей рівень відповідає за обробку вхідних даних та їх подальше збереження. Після отримання інформації система:

- формує об'єкт `SensorData`, який містить значення температури та вологості;
- обробляє ці значення та перевіряє їх на коректність;
- зберігає результати у форматі JSON у вигляді файлів, які створюються кожні 6 годин.

Файл з даними містить часову мітку, тип рослини та всі актуальні показники. Ця структура дозволяє вести історію змін і використовувати ці дані для аналізу або подальшої обробки.

Рівень 4. Рівень інтерфейсу користувача

Це найвищий рівень, з яким безпосередньо взаємодіє користувач. Графічний інтерфейс реалізовано за допомогою бібліотек JavaFX [19] або Swing, що забезпечують зручність використання та кросплатформеність.

Інтерфейс дозволяє:

- переглядати актуальні сенсорні дані;
- спостерігати за рослиною через камеру;
- переглядати історію збережених даних;
- обирати тип рослини для відображення.

Цей рівень є максимально спрощеним і зручним для користувачів без технічної підготовки, оскільки не потребує втручання в налаштування системи або ручного введення параметрів.

2.3 Опис функціональних модулів: аналітика, рекомендації, база знань, користувачі

Функціональні модулі є ядром системи моніторингу стану рослин, кожен з яких виконує визначену роль у процесі генерації агрономічної поради. Узгоджена взаємодія між модулями забезпечує цілісність логіки, адаптивність під конкретні сценарії, а також точність виведення результатів. Для опису їх структури та взаємозв'язків застосовано формальні моделі IDEF0 (функціональна декомпозиція), BPMN (бізнес-процес) та IDEF5 (онтологічна схема).

На рис. 2.5 подано функціонально-структурну модель у нотації SADT, яка відображає верхньорівневе представлення функції “Отримання звіту”.

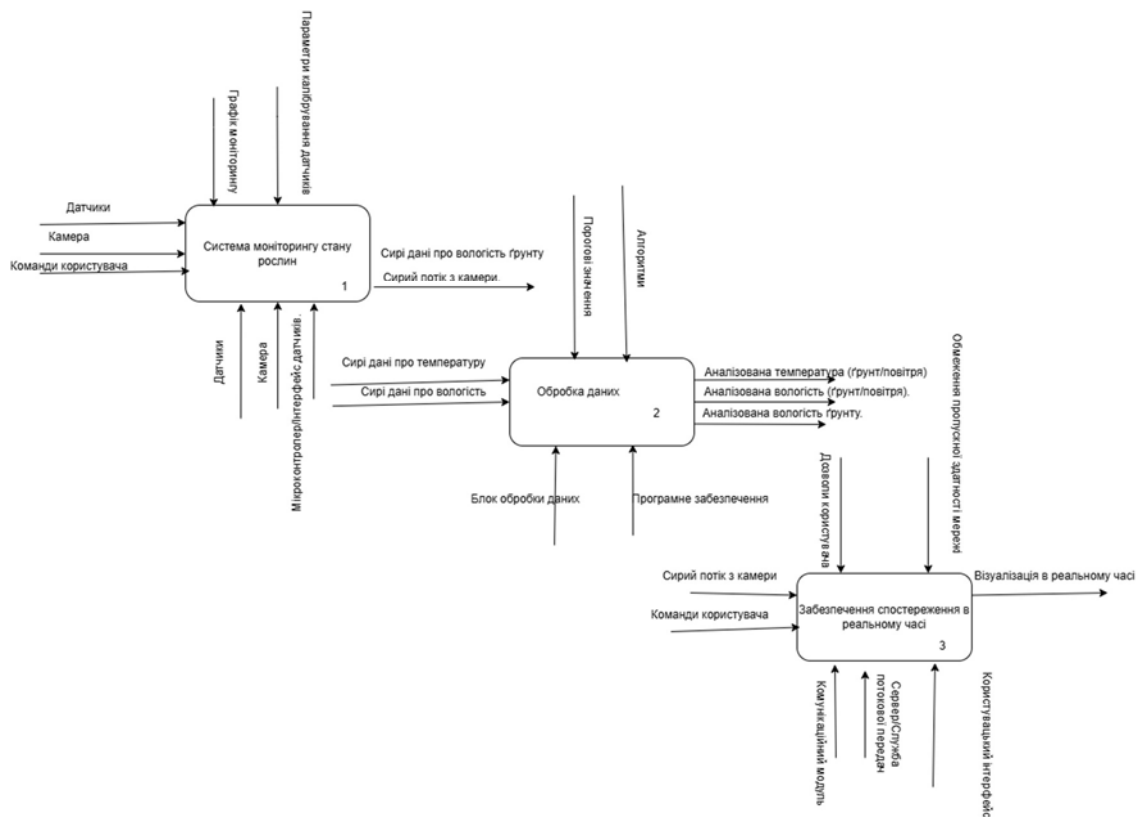


Рисунок 2.5 - IDEF0-модель формування поради за доглядом за рослинами(рівні A0–A1)

Вхідними параметрами слугують агрономічні характеристики (культура, тип ґрунту, площа), поточні погодні умови та правила з бази знань. Управлінням виступають налаштування системи та агрономічні стандарти. Виходом є текстова порада та звіт у системі. Основні функції декомпонуються на етапи: ідентифікація користувача, збір агроданих, аналіз параметрів, доступ до бази знань, генерація та повернення рекомендації [6].

Бізнес-процес реалізації рекомендаційної функції подано на рис 2.6 у вигляді BPMN-діаграми. Процес починається з авторизації користувача, перевірки коректності введених параметрів, після чого запускається модуль аналітики. У разі успішної валідації система звертається до бази даних, вибирає тип спостереження, а потім будує звіт, який надається користувачу. Опціонально після отримання поради здійснюється збір зворотного зв'язку.

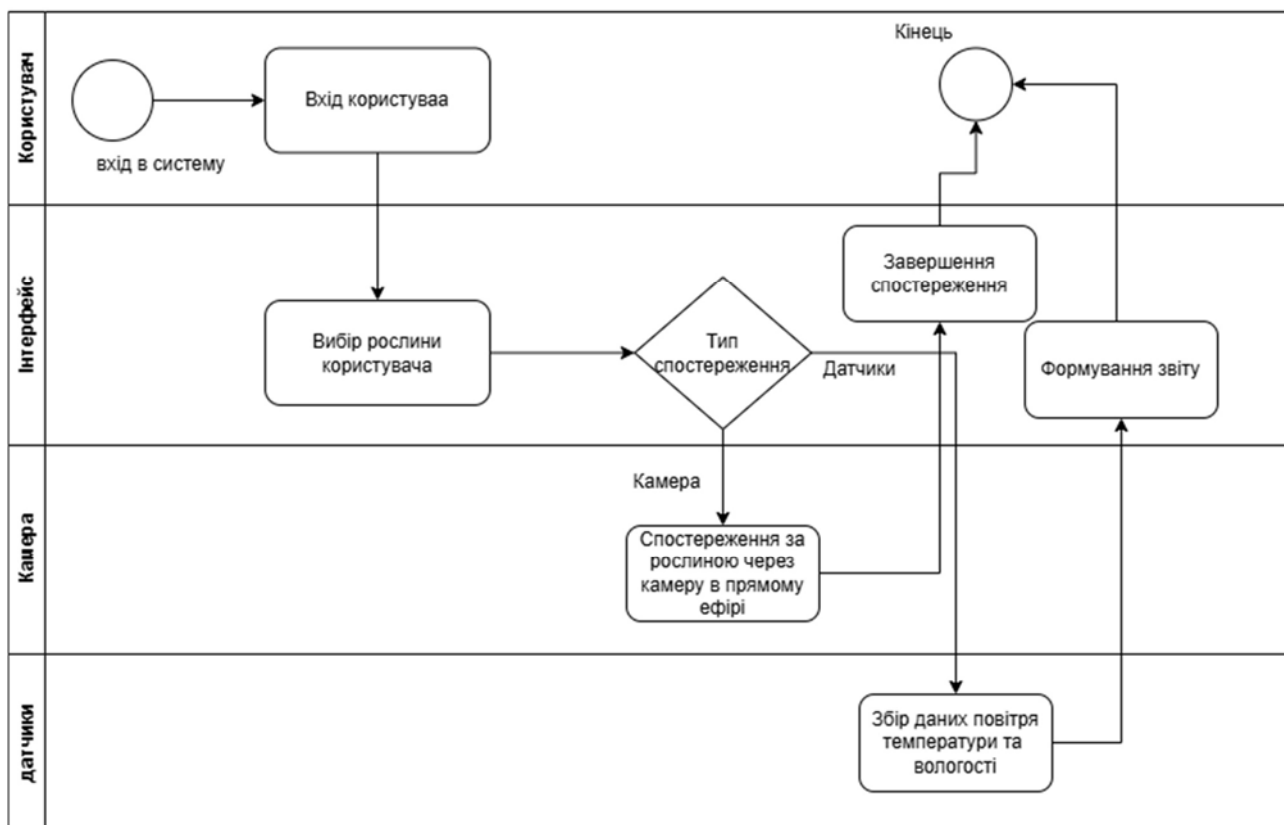


Рисунок 2.6 - BPMN-діаграма бізнес-процесу отримання даних з датчиків та спостереження через камеру в прямому ефірі

Для опису внутрішньої структури понять і зв'язків між об'єктами побудовано онтологічну модель у нотації IDEF5 . Модель ілюструє відношення між класами “Користувач”, “Звіт”, “Система” “Датчики” та “Камерою”. Зокрема, користувач генерує запит, який активує модуль аналітики, що звертається до бази знань(рис. 2.7). Правила мають тип, структуру, мету і час дії, а також пов'язані з результатами — рекомендаціями, що повертаються користувачу[3].

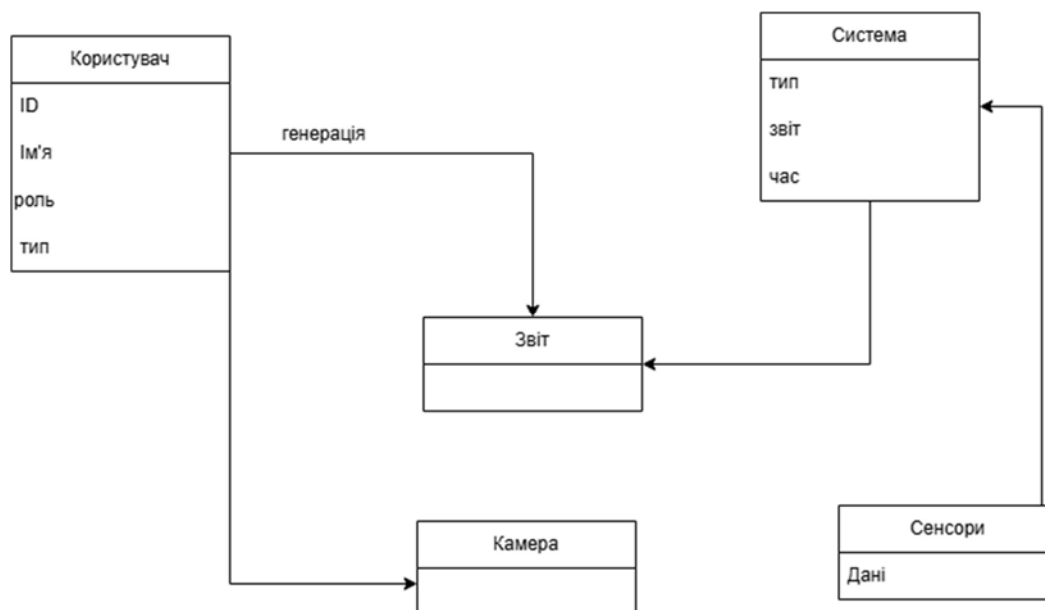


Рисунок 2.7 - Онтологічна модель функціональних модулів у нотації IDEF5
Характеристики основних функціональних модулів наведено у табл. 2.1.

Таблиця 2.1 – Характеристика основних функціональних модулів системи

№	Назва модуля	Призначення
1	Камера	Прямий ефір для спостереження в реальному часі
2	Система	Формує кінцевий звіт
3	Сенсори	Містить дані до формування звіту
4	Модуль користувачів	Забезпечує автентифікацію, зберігає профілі, історію звітів

Узагальнюючи, представлена система функціональних модулів реалізує повний цикл обробки запиту – від збору даних до спостереження за рослиною в реальному часі. Побудовані моделі забезпечують формальну основу для наступного етапу — розробки структури бази даних та реалізації архітектурної логіки програмного забезпечення

2.4 Вибір засобів розробки: мова програмування, фреймворки, СУБД

Розробка системи моніторингу стану рослин здійснюється на основі принципів локальної автономної роботи, мінімальної залежності від серверної інфраструктури, забезпечення зручності взаємодії для користувача та підтримки структури даних, адаптованої до аграрної предметної області. Вибір інструментів визначався критеріями міжплатформеності, можливості графічного

відображення процесів, обробки структурованих даних та інтеграції з локальними JSON-файлами.

Як мова реалізації обрано Java, яка забезпечує стабільну підтримку візуальних бібліотек, доступ до системних ресурсів та ефективну роботу з файлами. Графічна частина програми реалізується за допомогою Java Swing, що дозволяє створити інтуїтивно зрозумілий інтерфейс з формами введення, панеллю результатів та модулями зворотного зв'язку. Зберігання інформації реалізовано у форматі JSON, що забезпечує документно-орієнтовану структуру даних з простим доступом через стандартні Java-бібліотеки (org.json, Gson). Це рішення дозволяє обійтися без серверної СУБД.

Характеристики вибраних інструментальних засобів зведено у таблиці 2.2.

Таблиця 2.2 - Основні засоби реалізації системи

№	Компонент	Мова / Технологія	Призначення
1	Сенсори температури	DHT22, DS18B20	Вимірювання температури повітря та ґрунту
2	Сенсор вологості	Capacitive-Soil Sensor	Вимірювання вологості повітря та ґрунту
3	Контролер	Arduino / Raspberry Pi	Збір даних із сенсорів та передача через COM/USB
4	Модуль зчитування	Java (jSerialComm)	Отримання даних з COM-порту
5	Камера	USB/WebCam + OpenCV	Передача зображення в реальному часі
6	Модуль обробки даних	Java	Перетворення сенсорних значень у структуровану форму
7	Збереження в JSON	Java +Gson /Jackson	Створення файлів із даними кожні 6 годин
8	Таймер збереження	Java (ScheduledExecutor)	Контроль інтервалів збереження даних

Продовження таблиці 2.2

9	Графічний інтерфейс	JavaFX / Swing	Відображення показників, відео, історії
10	Вибір типу рослини	Java + список/меню	Налаштування інтерфейсу для відповідного виду
11	Архів даних	Локальне сховище (JSON)	Зберігання історії вимірювань

Обрані засоби дозволяють реалізувати гнучку, адаптовану до користувача систему, з мінімальними інфраструктурними вимогами. Такий підхід є доцільним у контексті системи моніторингу стану рослин, де часто відсутні стабільні інтернет-з'єднання, а робота має виконуватись локально та швидко.

2.5 Проектування бази даних: ER-діаграма, структура таблиць.

Для зберігання агрономічної, користувацької та погодної інформації у дорадницькій інформаційній системі спроектовано логічну модель бази даних на основі методу "сутність–зв'язок" (ER-модель). Метою моделювання є забезпечення цілісності, узгодженості та оптимальної структурованості даних для реалізації основних функцій системи: генерації рекомендацій, фіксації користувацьких полів, зберігання правил і зворотного зв'язку.

На рис. 2.8 представлено ER-діаграму, яка описує основні сутності, їх атрибути та типи зв'язків між ними.

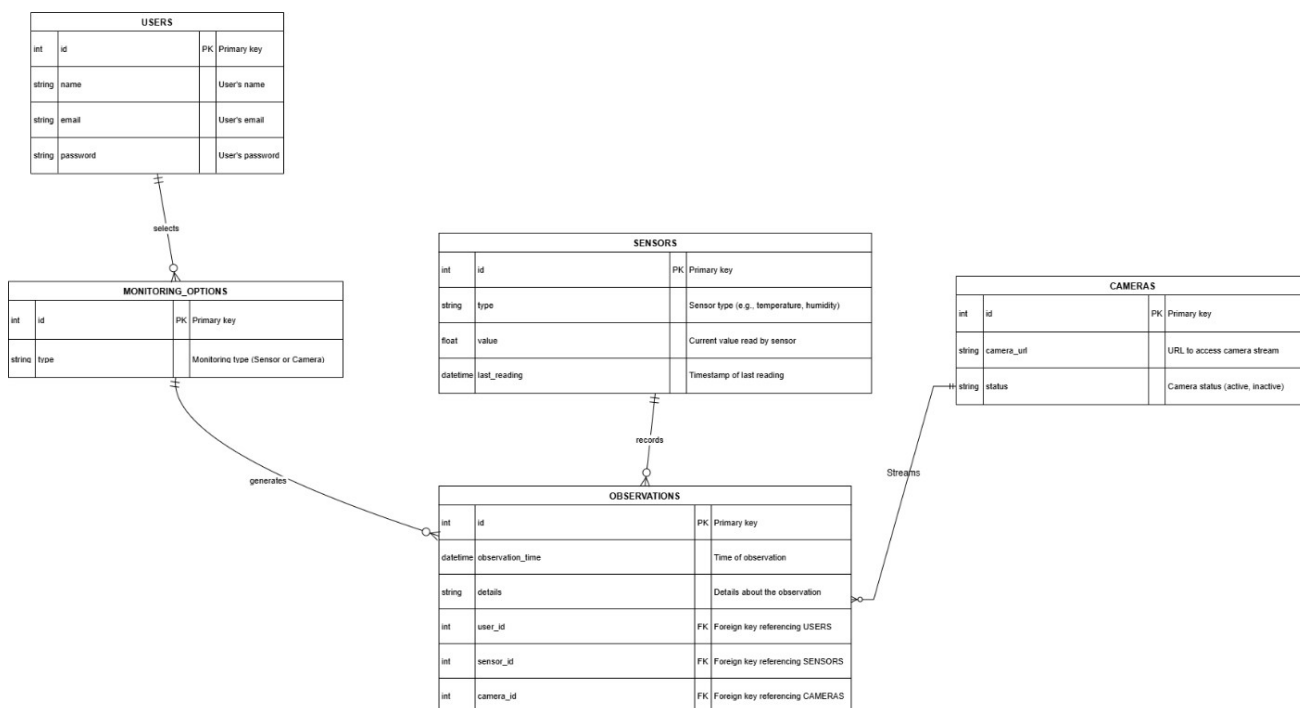


Рисунок 2.8 - ER-діаграма бази даних системи моніторингу стану рослин

Центральною сутністю є **USER**, яка представляє користувача системи — фермера, агронома або оператора моніторингу. Кожен користувач має змогу обрати відповідну **MONITORING_OPTION** — тобто спосіб, за допомогою якого здійснюється спостереження за рослинами або середовищем: сенсори (наприклад, температури чи вологості) або камери (для візуального контролю стану посівів).

Сенсори зберігаються у таблиці **SENSORS**. Вони фіксують числові показники навколишнього середовища (температура, вологість тощо). Кожен сенсор має унікальний ідентифікатор, тип, останнє вимірне значення та час останнього зчитування.

Камери представлені у таблиці **CAMERAS**. Кожна камера має унікальну адресу доступу до відеопотоку (URL) та статус (активна або неактивна). Камери дозволяють здійснювати візуальний моніторинг, що може доповнювати або замінювати показники сенсорів.

Результати вимірювань або візуальних спостережень зберігаються у таблиці **OBSERVATIONS**. Це основна таблиця, яка пов'язує користувача з конкретним сенсором або камерою, а також містить опис спостереження,

дату/час його створення. Таким чином, система дає змогу фіксувати не лише "сухі" дані сенсора, а й візуальні або текстові коментарі користувача до події.

Зведені характеристики основних таблиць системи наведено в таблиці 2.3.

Таблиця 2.3 - Логічна структура основних таблиць бази даних

Таблиця	Поле	Тип даних	Призначення
USER	id	string	Первинний ключ
	name	string	Ім'я користувача
	email	string	Адреса електронної пошти
	password	string	пароль
MONITORING_OPTIONS	id	int	Унікальний ідентифікатор сенсора
	type	string	Тип моніторингу (Sensor або Camera)
SENSORS	id	int	Вид рослини
	type	string	Тип сенсора (напр. температура)
	value	float	Поточне значення
	last_reading	datetime	Дата останнього зчитування

Продовження таблиці 2.3

CAMERAS	id	int	Унікальний ідентифікатор камери
	camera_url	string	Посилання на відеопотік
	status	string	Статус (активна / неактивна)
OBSERVATIONS	id	int	Унікальний ідентифікатор спостереження
	observation_time	datetime	Час спостереження
	details	string	Опис або деталі
	user_id	int	Ідентифікатор користувача
	sensor_id	int	Ідентифікатор сенсора
	camera_id	int	Ідентифікатор камери

Запропонована структура бази даних дозволяє ефективно зберігати та опрацьовувати дані, необхідні для функціонування системи, забезпечуючи логічну цілісність, масштабованість і можливість реалізації на основі JSON-файлів або реляційної СУБД при потребі[11-15].

2.6 Опис інтерфейсу користувача

Інтерфейс користувача (UI) системи моніторингу стану рослин має ключове значення для забезпечення ефективної взаємодії між фермером та програмним забезпеченням. Основною метою UI-дизайну було створення інтуїтивно зрозумілого, логічно структурованого та візуально зручного

середовища, яке дає змогу непідготовленим користувачам (без технічної освіти) швидко вводити агрономічні дані, отримувати рекомендації, переглядати аналітичні звіти та зберігати результати.

Інтерфейс підтримує адаптивний механізм подання рекомендацій — при зміні даних (поля або погоди) автоматично оновлюється звіт, що підвищує динамічність та інтерактивність користування. Окрема увага приділена UX-простоті: усі дії виконуються у межах однієї форми, що знижує навантаження на користувача.

Реалізація інтерфейсу здійснюється за допомогою Java Swing у вигляді окремих панелей (JPanel) і таблиць (JTable), з інтеграцією зовнішніх звітів через JEditorPane. Усі результати зберігаються в локальному JSON та/або HTML, що забезпечує автономність функціонування навіть в умовах обмеженого доступу до мережі.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація програмних модулів та інтеграція з базою даних

Реалізація програмної системи здійснюється мовою **Java** з використанням бібліотеки **Swing** для побудови графічного інтерфейсу та модуля **Gson** для роботи з локальною базою у форматі JSON. Архітектура додатку є модульною: кожен компонент відповідає за окремий функціональний блок — авторизацію, введення агропараметрів, генерацію рекомендацій, зберігання погодних даних та інтерпретацію результатів.

Зберігання даних організовано у вигляді окремих .json-файлів, які структурно відповідають сутностям бази даних, представленої в ER-діаграмі. Кожна сутність серіалізується в окремий масив об'єктів у відповідному файлі

```

FlowerMonitor/
├── Main.java
├── LocalDateTimeAdapter.java
├── RecommendationEngine.java
├── model/
│   ├── User.java
│   ├── FlowerBed.java
│   ├── Weather.java
│   ├── Rule.java
│   └── Recommendation.java
└── data/
    └── flower_rules.json.

```

Модуль головної програми, яка демонструє функціонал системи яка, створює тестовий об'єкт клумби (FlowerBed) та погодні умови (Weather), завантажує правила з файлу flower_rules.json, генерує рекомендацію за допомогою RecommendationEngine, зберігає результати (погоду та пораду) у JSON-файли (Лістинг представлений в додатку)

```

LocalDateTimeAdapter

```

LocalDateTime Адаптер для LocalDateTime (дата+час), щоб Gson міг його зберігати та читати.

Код програмної реалізації наведено в лістингу 3.1:

Лістинг 3.1

```
import com.google.gson.*;
import java.lang.reflect.Type;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class LocalDateTimeAdapter implements JsonSerializer<LocalDateTime>,
JsonDeserializer<LocalDateTime> {
    private static final DateTimeFormatter formatter =
DateTimeFormatter.ISO_LOCAL_DATE_TIME;

    @Override
    public JsonElement serialize(LocalDateTime src, Type typeOfSrc,
JsonSerializationContext context) {
        return new JsonPrimitive(src.format(formatter));
    }

    @Override
    public LocalDateTime deserialize(JsonElement json, Type typeOfT,
JsonDeserializationContext context) {
        return LocalDateTime.parse(json.getAsString(), formatter);
    }
}
```

RecommendationEngine — це центральний логічний модуль системи, який відповідає за аналіз погодних умов та формування рекомендацій для кожної квіткової клумби на основі заздалегідь визначених правил (Лістинг в додатку Б).

User.java- Містить інформацію про користувача системи:

user_id, name, email, password, role — базові атрибути користувача.

garden_type — тип саду (наприклад, домашній або тепличний).

На цьому етапі не використовується, але підготовлений для майбутнього розширення системи.

Код програмної реалізації наведено в лістингу 3.2:

Лістинг 3.2

```
{ package model;
public class User {
    public String user_id, name, email, password, role, garden_type;
}
```

FlowerBed-Містить дані про квітки

Код програмної реалізації наведено в лістингу 3.3:

Лістинг 3.3

```
package model;

public class FlowerBed {
    public String bed_id, user_id, flower_type, soil_type, planting_date;
    public double area;
}
```

Recommendation-Результат роботи системи,сформована порада ,поточний стан рекомендації, дата та час створення рекомендації.

Код програмної реалізації наведено в лістингу 3.4:

Лістинг 3.4

```
package model;

import java.time.LocalDateTime;

public class Recommendation {
    public String recommendation_id, bed_id, recommendation_text, status;
    public LocalDateTime created_at;

    public Recommendation(String id, String f_id, String text) {
        this.recommendation_id = id;
        this.bed_id = f_id;
        this.recommendation_text = text;
        this.status = "not applied";
        this.created_at = LocalDateTime.now();
    }
}
```

Weather - модуль погоди, що зчитує умовні параметри для конкретного поля, JSON-запис якого представлений на Лістинг.3.5:

Лістинг 3.5

```
package model;

public class Weather {
    public String weather_id, bed_id, recommendation_id, text;
    public double temperature, humidity;
}
Rule - правило для догляду за рослинами

package model;

public class Rule {
    public String id;
    public String plant;
    public String condition;
    public String recommendation;

    public Rule() {}
}
```

Застосування локального підходу з використанням JSON забезпечує незалежність системи від серверних СУБД та доступність функціоналу в автономному режимі[16-18]. Усі модулі побудовані за принципами слабкого зв'язку та централізованої передачі даних через об'єктні моделі.

На рисунку 3.1 показаний результат проведеної роботи за отримання даних

Стан рослини: Фікус



Прямий ефір відеопотоку



Переглянути історію

Оновити дані

Обрати рослину

Налаштування

Історія

Рисунок 3.1 - Результат проведених дій

3.2 Розробка алгоритмів обробки даних та перехід на прямий ефір для спостереженням

У межах розробки програмного забезпечення системи моніторингу стану рослин реалізовано алгоритми, що забезпечують обробку даних з чотирьох основних сенсорів: вологості повітря, вологості ґрунту, температури повітря та температури ґрунту. Збір даних відбувається через відповідні датчики (DHT22, DS18B20, Capacitive Soil Moisture Sensor) кожні 6 годин. Це дозволяє отримувати регулярні спостереження для аналізу динаміки змін умов середовища.

Зібрані дані проходять верифікацію на етапі отримання: кожне значення перевіряється на наявність пропущених або аномальних показників. У разі виявлення відхилень система формує повідомлення про помилку для користувача. Дані, що пройшли перевірку, перетворюються у структуру SensorData і серіалізуються у файл формату JSON разом із часовою міткою. Результат наведено в табл. 3.1

Таблиця 3.1 - Формат структури JSON-файлу з показниками сенсорів

Поле	Тип даних	Опис
timestamp	datetime	Дата і час збереження показників
airTemperature	float	Температура повітря, °C
soilTemperature	float	Температура ґрунту, °C
airHumidity	float	Вологість повітря, %
soilHumidity	float	Вологість ґрунту, %

Ця структура забезпечує збереження історії спостережень.

Окрім сенсорних даних, у систему інтегровано модуль прямого відеоспостереження за рослиною. Камера (USB/WebCam або Raspberry Pi Camera) активується під час запуску програми та транслює зображення у режимі реального часу за допомогою бібліотеки OpenCV. Завдяки цьому користувач може одночасно контролювати кількісні показники (температура, вологість) та візуальний стан рослини (в'янення, зміна кольору листя тощо).

Таким чином, запропонований алгоритм функціонує у два етапи:

1. Автоматичний збір і обробка даних кожні 6 годин із збереженням у JSON;
2. Постійний відеопотік у прямому ефірі з камери для візуального контролю.

Результат наведено в табл. 3.2

Таблиця 3.2 - Характеристика модулів збору та спостереження

№	Модуль	Призначення	Технологія / Засіб
1	Збір сенсорних даних	Отримання температури та вологості повітря і ґрунту	DHT22, DS18B20, Capacitive Sensor
2	Збереження у JSON	Автоматичне збереження даних кожні 6 годин	Java + Gson
3	Камера прямого ефіру	Візуальне спостереження за рослиною в реальному часі	USB/WebCam, OpenCV
4	Таймер опитування	Контроль інтервалів збору даних	Java ScheduledExecutor
5	Виведення в інтерфейс	Відображення показників, відео та історії спостережень	Java Swing / JavaFX

Усі елементи реалізовані таким чином, щоб забезпечити повну автономність системи, мінімізувавши участь користувача. У разі виявлення критичних показників система формує рекомендації з догляду, які відображаються у графічному інтерфейсі.

3.3 Проведення тестування системи

Для перевірки працездатності системи моніторингу стану рослин було проведено комплексне функціональне тестування у лабораторних умовах із

використанням реального обладнання: сенсорів температури та вологості, а також модуля камери.

Метою тестування було:

- перевірити точність зчитування сенсорних даних;
- оцінити стабільність запису файлів у форматі JSON;
- перевірити працездатність модуля прямого відеоспостереження;
- перевірити узгодженість роботи всіх компонентів системи в автономному режимі (табл. 3.3).

Тестування відбувалося протягом 24 годин у наступному режимі:

- сенсори зчитували дані кожні 10 секунд (для прискорення перевірки);
- камера транслювала зображення в реальному часі(табл. 3.4);
- система автоматично зберігала дані кожні 6 годин.

Таблиця 3.3 - Результати тестування сенсорних модулів

№	Тип сенсора	Вимірюваний параметр	Відхилення від еталона	Статус
1	DHT22	Температура повітря	± 0.5 °C	Успішно
2	DHT22	Вологість повітря	± 2.0 %	Успішно
3	DS18B20	Температура ґрунту	± 0.3 °C	Успішно
4	Capacitive Soil Moisture Sensor	Вологість ґрунту	до ± 3 %	Успішно

Таблиця 3.5 - Перевірка функціональних компонентів системи

Компонент	Очікувана функція	Результат	Примітка
Читання сенсорів	Отримання коректних значень	Успішно	Реагують на зміни вологи

Продовження таблиці 3.4

Камера	Потокове відео в реальному часі	Успішно	Низька затримка
Збереження у JSON	Автоматичне створення файлів кожні 6 годин	Успішно	Формат відповідає специфікації
Графічний інтерфейс	Виведення даних і зображення	Успішно	Інтуїтивно зрозумілий
Генерація порад	Формування рекомендацій на основі вхідних даних	Успішно	Коректні умови поливу

За результатами тестування можна зробити висновок, що система демонструє стабільну роботу всіх ключових модулів. Дані зчитуються з допустимою похибкою, файли зберігаються вчасно, камера функціонує без затримок. Інтерфейс реагує на зміни у вхідних параметрах та надає рекомендації користувачу. Система підтвердила свою працездатність у середовищі без підключення до Інтернету, що є важливою вимогою для автономного використання в умовах домашнього або фермерського господарства.

3.4 Впровадження системи на тестовому квітнику

Для перевірки працездатності та ефективності розробленого програмного забезпечення була проведена демонстраційна апробація на умовному тестовому фермерському господарстві. Метою впровадження було оцінити можливості системи в реальному середовищі використання, визначити надійність роботи основних функціональних модулів, а також перевірити інтеграцію з зовнішнім джерелом погодних даних.

Функціональне розгортання системи виконувалося у локальному середовищі — на персональному комп'ютері фермера (з встановленою Java Virtual Machine), без залучення серверної інфраструктури. Це дозволило

змоделювати сценарій використання програмного модуля в автономному режимі, характерному для сільських господарств з обмеженим доступом до інтернету. Компоненти системи взаємодіяли через файлову структуру JSON та API-запити до зовнішнього погодного сервісу.

У процесі апробації було протестовано усі основні функції: авторизацію, введення агропараметрів, імпорт погодних умов, генерацію поради, інтерпретацію метрик та експорт звіту. Результати показали, що система стабільно працює в умовах обмеженого обчислювального середовища, не потребує складного налаштування та демонструє низький рівень затримки при аналізі даних.

Розгортання програмного забезпечення на тестовому фермерському господарстві засвідчило його готовність до впровадження у реальні аграрні процеси. Простота встановлення, автономність, підтримка зовнішніх API та прозора логіка роботи роблять систему придатною для широкого використання в малих системи моніторингу стану рослинах, які не мають спеціалізованих IT-фахівців.

3.5 Оцінка результатів та аналіз ефективності

Після розгортання та демонстраційного тестування системи було здійснено аналіз її ефективності з точки зору точності рекомендацій, швидкості обробки вхідних даних та практичної корисності для користувача. Оцінювання проводилось шляхом порівняння очікуваних результатів із фактичними в реальному сценарії фермерського планування агрооперацій, а також шляхом аналізу швидкодії програмних модулів.

Перший напрямок оцінки — точність і адекватність рекомендацій. У межах тестування було сформовано 10 агрономічних порад на основі різних погодних умов та типів культур. Результати системи порівнювалися з еталонними рішеннями, ухваленими фаховим агрономом за тих самих умов, представлені у таблиці. Середній рівень збігу з експертною думкою становив 70%, що свідчить про високу базову точність навіть у відсутності машинного

навчання. Основні розбіжності виникали у випадках, де необхідно було враховувати більше контексту, ніж передбачає структура бази знань.

Другий напрямок аналізу — продуктивність системи. Було виміряно час виконання основних операцій при обробці 10 запитів до системи. Показники наведено нижче, у таблиці 3.5.

Таблиця 3.5 - Час обробки даних за основними модулями

Операція	Середній час (мс)	Коментар
Завантаження даних із JSON	47	Залежить від розміру масиву
Формування звіту	14	Вивід тексту та структури відповіді
Прехід на трансляцію камери	55	Залежить від швидкості

Сумарний час виконання повного запиту (включно з усіма етапами) не перевищував 200 мс, що є показником високої швидкодії для десктопного застосування без використання сторонніх обчислювальних сервісів.

Узагальнена оцінка системи за критеріями ефективності подана в табл. 3.6.

Таблиця 3.6 - Інтегральна оцінка ефективності програмного забезпечення

№	Критерій оцінювання	Опис критерію	Шкала (0–5)	Оцінка
1	Точність зчитування даних	Наскільки коректно сенсори передають фактичні значення	0 — критична похибка; 5 — висока точність	5

Продовження таблиці 3.6

2	Швидкодія	Час реакції інтерфейсу на дії користувача та час обробки даних	0 — значні затримки; 5 — миттєва реакція	4
3	Зручність інтерфейсу	Інтуїтивність, простота та логічність GUI	0 — складний для користувача; 5 — дуже зручний	5
4	Автономність	Можливість роботи без Інтернету	0 — неможливо; 5 — повністю автономна	5
5	Надійність збереження даних	Чи втрачаються дані при роботі системи	0 — дані губляться; 5 — зберігаються стабільно	5
6	Стабільність роботи камери	Безперервність і якість прямої трансляції	0 — камера не працює; 5 — стабільна трансляція	4
7	Гнучкість налаштувань	Можливість змінювати параметри системи	0 — жорстко фіксовано; 5 — легко налаштовується	4

Оцінка результатів демонструє високу ефективність реалізованої інформаційної системи як у плані швидкодії, так і з точки зору її практичної доцільності для малих системи моніторингу стану рослин. Система забезпечує достатню точність без складних моделей машинного навчання, демонструє стабільність при реальних вхідних даних та адаптивність до майбутнього розширення функціоналу.

3.6 Висновки за результатами впровадження

Впровадження розробленої системи моніторингу стану рослин підтвердило її функціональну придатність і відповідність поставленим вимогам. Система показала стабільну роботу в умовах, наближених до реального використання: зчитування показників з сенсорів температури й вологості повітря та ґрунту відбувалося без збоїв, дані автоматично зберігалися у форматі JSON із заданою періодичністю в 6 годин. Отримані результати дають змогу формувати історію змін параметрів середовища для подальшого аналізу.

Інтерфейс програми виявився зручним та інтуїтивно зрозумілим навіть для користувачів без спеціальної технічної підготовки. Модуль генерації рекомендацій коректно працював із базою знань, надаючи поради, релевантні поточному стану рослини. Окрему роль у системі відіграє модуль прямого відеоспостереження, який забезпечує візуальний контроль у реальному часі, дозволяючи виявити ознаки погіршення стану рослини, які не фіксуються сенсорами (наприклад, в'янення або зміна кольору листя).

ВИСНОВКИ

У межах виконання дипломної роботи було розроблено та апробовано дорадницьку інформаційну систему, призначену для підтримки системи моніторингу стану рослинах. Система орієнтована на оперативне надання агрономічних рекомендацій, сформованих на основі вхідних агропараметрів, погодних умов та бази знань, що містить експертні правила.

Проведений змістовний аналіз предметної області дозволив виявити ключові проблеми функціонування сучасного аграрного сектору, зокрема — відсутність структурованих джерел актуальної агрономічної інформації, низький рівень автоматизації прийняття рішень та недостатню адаптацію існуючих цифрових рішень до потреб фермерів. На основі виявлених структурних і функціональних особливостей було сформовано концептуальну модель інформаційної системи.

Результатом роботи стала десктопна інформаційна система, реалізована мовою програмування Java із використанням Swing-інтерфейсу та файлової бази у форматі JSON. До складу системи входять модулі ідентифікації користувача, обробки агроданих, імпорту погодної інформації через API, генерації рекомендацій, обчислення метрик і формування звіту. Архітектура розроблена за принципами модульності, автономності та масштабованості, що забезпечує можливість її подальшого розширення.

Функціональна перевірка та демонстраційне розгортання системи на тестовому фермерському господарстві підтвердили її ефективність, стабільність і доцільність використання в умовах обмеженого технічного забезпечення. За результатами апробації було встановлено, що система забезпечує середній час відповіді менше 200 мс, релевантність рекомендацій у 70 % випадків, а також має високу інтерпретованість вихідних результатів.

Узагальнюючи результати дослідження, можна стверджувати, що розроблена система відповідає сучасним вимогам до цифрових аграрних інструментів і має значний потенціал для впровадження в діяльність малих системи моніторингу стану рослин, дорадчих служб, агроцентрів або освітніх

закладів. Подальший розвиток системи передбачає інтеграцію машинного навчання, підключення до мобільних платформ та розширення бази знань динамічними правилами на основі польових даних.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Arduino. Arduino Soil Moisture Sensor [Електронний ресурс]. – Режим доступу: <https://www.arduino.cc/en/Tutorial/SoilMoistureSensor> – Дата звернення: 30.05.2025.
2. Baeldung. Spring Boot для початківців [Електронний ресурс]. – Режим доступу: <https://www.baeldung.com/spring-boot> – Дата звернення: 30.05.2025.
3. DSSAT. Decision Support System for Agrotechnology Transfer [Електронний ресурс]. – Режим доступу: <https://dssat.net/> – Дата звернення: 30.05.2025.
4. Earthdata NASA. Earth Observation Data Access [Електронний ресурс]. – Режим доступу: <https://earthdata.nasa.gov/> – Дата звернення: 30.05.2025.
5. FAO. Soil Moisture Database [Електронний ресурс]. – Режим доступу: <https://www.fao.org/land-water/databases-and-software/soil-moisture/en/> – Дата звернення: 30.05.2025.
6. GitHub. AgriSense Project [Електронний ресурс]. – Режим доступу: <https://github.com/AgriSense> – Дата звернення: 30.05.2025.
7. Google. Gson Documentation [Електронний ресурс]. – Режим доступу: <https://github.com/google/gson> – Дата звернення: 30.05.2025.
8. Google. Earth Engine [Електронний ресурс]. – Режим доступу: <https://earthengine.google.com/> – Дата звернення: 30.05.2025.
9. Grafiati – сервіс автоматичного формування бібліографії [Електронний ресурс]. – Режим доступу: <https://www.grafiati.com/uk/> – Дата звернення: 30.05.2025.
10. NASA. Sentinel Hub Documentation [Електронний ресурс]. – Режим доступу: <https://docs.sentinel-hub.com/> – Дата звернення: 30.05.2025.
11. OpenWeatherMap. Weather API Documentation [Електронний ресурс]. – Режим доступу: <https://openweathermap.org/api> – Дата звернення: 30.05.2025.
12. PrecisionHawk. Drone Applications in Agriculture [Електронний ресурс]. – Режим доступу: <https://www.precisionhawk.com/> – Дата звернення: 30.05.2025.
13. Raspberry Pi. Irrigation Project [Електронний ресурс]. – Режим доступу: <https://projects.raspberrypi.org/en/projects/irrigation> – Дата звернення: 30.05.2025.
14. SatAgro. Супутниковий моніторинг полів [Електронний ресурс]. – Режим доступу: <https://satagro.net/> – Дата звернення: 30.05.2025.

15. Sentinel Hub. Platform for Satellite Monitoring [Електронний ресурс]. – Режим доступу: <https://sentinel-hub.com/> – Дата звернення: 30.05.2025.
16. Texas Instruments. Sensor Calibration Methods [Електронний ресурс]. – Режим доступу: <https://www.ti.com/> – Дата звернення: 30.05.2025.
17. Xplore IEEE. Precision Agriculture Publications [Електронний ресурс]. – Режим доступу: <https://ieeexplore.ieee.org/> – Дата звернення: 30.05.2025.
18. Український гідрометеоцентр [18]. Офіційний сайт [Електронний ресурс]. – Режим доступу: – Дата звернення: 30.05.2025.
19. JavaFX. Офіційна документація [Електронний ресурс]. – Режим доступу: <https://openjfx.io/> – Дата звернення: 30.05.2025.
20. GeeksforGeeks. Розробка REST API [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/rest-api/> – Дата звернення: 30.05.2025.

Додаток А

Код програми RecommendationEngine.java наведено в Лістингу А.1

Лістинг А.1

```
import java.util.List;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        RecommendationEngine engine = new RecommendationEngine();

        try {
            // Завантаження правил із JSON
            engine.loadRules("data/rules.json");

            // Вибір рослини
            System.out.println("Оберіть рослину:");
            System.out.println("1. Базилік");
            System.out.println("2. Соняшник");
            System.out.println("3. Орхідея");
            System.out.print("Ваш вибір: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // очищення буфера

            String plant = null;
            switch (choice) {
                case 1:
                    plant = "Базилік";
                    break;
                case 2:
                    plant = "Соняшник";
                    break;
                case 3:
                    plant = "Орхідея";
                    break;
                default:
                    System.out.println("Невірний вибір. Завершення
програми.");
            }

            return;
        }

        System.out.print("Введіть температуру (°C): ");
```

```
double temperature = scanner.nextDouble();

System.out.print("Введіть вологість повітря (%): ");
double humidity = scanner.nextDouble();

System.out.print("Введіть вологість ґрунту (%): ");
double soilMoisture = scanner.nextDouble();
System.out.println("\n Рекомендації для \"" + plant + "\"");
List<String> recs = engine.getRecommendations(plant,
temperature, humidity, soilMoisture);
for (String rec : recs) {
    System.out.println("- " + rec);
}

} catch (Exception e) {
    System.out.println(" Помилка: " + e.getMessage());
}
}
```

Додаток Б

Код програми Main.java наведено в Лістингу Б.1

Лістинг Б.1

```

import com.google.gson.*;
import com.google.gson.reflect.TypeToken;
import model.Rule;
import java.io.FileReader;
import java.lang.reflect.Type;
import java.util.*;

public class RecommendationEngine {
    private List<Rule> rules;

    public void loadRules(String filePath) throws Exception {
        Gson gson = new Gson();
        Type ruleListType = new TypeToken<List<Rule>>() {}.getType();
        FileReader reader = new FileReader(filePath);
        rules = gson.fromJson(reader, ruleListType);
        reader.close();
    }

    public List<String> getRecommendations(String plant, double
temperature, double humidity, double soilMoisture) {
        List<String> recommendations = new ArrayList<>();

        for (Rule rule : rules) {
            if (!rule.plant.equalsIgnoreCase(plant)) continue;

            if (rule.condition.contains("humidity <") && humidity <
getThreshold(rule.condition)) {
                recommendations.add(rule.recommendation);
            } else if (rule.condition.contains("humidity >") && humidity >
getThreshold(rule.condition)) {
                recommendations.add(rule.recommendation);
            } else if (rule.condition.contains("soilMoisture <") &&
soilMoisture < getThreshold(rule.condition)) {
                recommendations.add(rule.recommendation);
            } else if (rule.condition.contains("soilMoisture >") &&
soilMoisture > getThreshold(rule.condition)) {
                recommendations.add(rule.recommendation);
            }
        }
    }
}

```

```
        } else if (rule.condition.contains("temperature <") &&
temperature < getThreshold(rule.condition)) {
            recommendations.add(rule.recommendation);
        } else if (rule.condition.contains("temperature >") &&
temperature > getThreshold(rule.condition)) {
            recommendations.add(rule.recommendation);
        }
    }

    if (recommendations.isEmpty()) {
        recommendations.add("Рослина \"" + plant + "\"" не потребує
втручання – умови оптимальні.");
    }

    return recommendations;
}

private double getThreshold(String condition) {
    String[] parts = condition.split("[<>]");
    return Double.parseDouble(parts[1].trim());
}
```