

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри інформаційних систем і
технологій

_____ професор, к.е.н., М.З. Швиденко
підпис ініціали та прізвище

_____ 202_ р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему

“Дорадницька інформаційна система фермерського господарства”

Спеціальність 126 “Інформаційні системи та технології”

Гарант освітньої програми

_____ к.е.н., доцент
(науковий ступінь та вчене звання)

_____ (підпис)

_____ Мокрієв Максим Володимирович
(ПІБ)

Керівник кваліфікаційної роботи

_____ д.т.н., професор
(науковий ступінь та вчене звання)

_____ (підпис)

_____ Смолій Вікторія Миколаївна
(ПІБ)

Виконав

_____ (підпис)

_____ Михайлюк Владислав Володимирович
(ПІБ)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Завідувач кафедри інформаційних систем і
технологій

_____ професор, к.е.н., М.З. Швиденко
підпис ініціали та прізвище
_____ 202_ р.

ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи
студенту(ці) Михайлюку Владиславу Володимировичу
Спеціальності 126 “Інформаційні системи та технології”

1. Тема роботи: “Дорадницька інформаційна система фермерського господарства ”
Затверджена наказом ректора від 16.12.2024 р. № 2245-С
2. Термін подання завершеної роботи на кафедру -
3. Вихідні данні: опис програмного забезпечення
4. Перелік питань, що розглядаються:
 1. Аналіз предметної області.
 2. Вибір та обґрунтування засобів для розробки системи.
 3. Проектування інформаційної системи.
 4. Висновки.
5. Календарний план

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	16 грудня 2024	Успішно
2	Аналіз предметної області, модельовання предметної області	Лютий 2025	Успішно
3	Поставновка завдання	Лютий 2025	Успішно
4	Розробка та тестування системи	Березень 2025	Успішно
5	Оформлення записки	Травень 2025	Успішно
6	Попередній захист	10 червня 2025	Успішно
7	Захист	16-17 червня 2025	Успішно

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

Завдання прийняв до виконання _____ / В. В. Михайлюк
Підпис ПІБ

Дата отримання завдання 16.12.2024

РЕФЕРАТ

Тема бакалаврської кваліфікаційної роботи: “Дорадницька інформаційна система фермерського господарства”.

Автор роботи: Михайлюк Владислав Володимирович.

Керівник роботи: Смолій Вікторія Миколаївна.

Пояснювальна записка: 62 с., 20 рис., 15 табл., 1 дод., 32 джерел.

Графічна частина: 11 презентаційних слайдів.

ДОРАДНИЦЬКА ІНФОРМАЦІЙНА СИСТЕМА ФЕРМЕРСЬКОГО
ГОСПОДАРСТВА.

Метою даної роботи є створення автоматизованої дорадницької інформаційної системи, призначеної для підтримки прийняття рішень у фермерських господарствах. Система має забезпечувати збір, обробку та аналіз агрономічних даних із різних джерел, використання формалізованих знань і аналітичних алгоритмів для генерації персоналізованих агротехнологічних рекомендацій. Запропоноване рішення спрямоване на подолання обмежень традиційного консультування, таких як низька масштабованість, затримки в оновленні інформації та відсутність адаптації до конкретних умов господарства. Розроблена система повинна бути доступною, зручною у використанні та здатною працювати в реальному часі, щоб підвищити ефективність агровиробництва, зменшити ризики та покращити рівень інформаційної підтримки фермерів.

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.....	11
1.3 Визначення вимог до інструментальних засобів та розробка технічного завдання	15
1.4 Висновок до першого розділу.....	19
2 ПРОЄКТУВАННЯ ДОРАДНИЦЬКОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	
21	
2.1 Моделювання предметної області.....	21
2.2 Архітектура програмного забезпечення.....	24
2.3 Опис функціональних модулів: аналітика, рекомендації, база знань, користувачі	27
2.4 Вибір засобів розробки: мова програмування, фреймворки, СУБД	30
2.5 Проєктування бази даних: ER-діаграма, структура таблиць.....	32
2.6 Опис інтерфейсу користувача	34
2.7 Висновок до другого розділу	36
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....	38
3.1 Реалізація програмних модулів та інтеграція з базою даних	38
3.2 Розробка алгоритмів обробки даних та генерації порад.....	41
3.3 Проведення тестування системи	45

3.4 Впровадження системи на тестовому фермерському господарстві	48
3.5 Оцінка результатів та аналіз ефективності	50
3.6 Висновки за результатами впровадження	52
3.7 Висновок до третього розділу	53
ВИСНОВКИ.....	55
СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
ДОДАТОК А.....	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

Аграрне дорадництво є одним із ключових елементів ефективного функціонування сільського господарства, оскільки забезпечує виробників науково обґрунтованими рекомендаціями щодо вибору культур, агротехнологій, управління ресурсами та запобігання ризикам. Наявність своєчасної та достовірної інформації безпосередньо впливає на продуктивність фермерських господарств і їхню здатність адаптуватися до змін природних, економічних або нормативних умов [1], [2].

Водночас традиційні засоби консультування, зокрема індивідуальні зустрічі з дорадниками, друковані бюлетені чи телефонне консультування, втрачають ефективність через обмежену масштабованість, відсутність персоналізації та затримку в оновленні знань. Ці проблеми особливо актуальні для малих та середніх фермерських господарств, які не мають постійного доступу до фахових джерел інформації або ресурсів для впровадження високоякісних комерційних платформ [3], [4].

Підвищити ефективність консультування в агросекторі дозволяє впровадження спеціалізованих інформаційних систем, здатних акумулювати агрономічні знання, обробляти великі обсяги даних і надавати персоналізовані рекомендації в автоматизованому режимі. Створення таких систем вимагає поєднання галузевої експертизи, сучасних методів аналізу даних та зручних інтерфейсів для користувача [5], [6].

Метою даної роботи є розробка дорадницької інформаційної системи для фермерського господарства, яка забезпечує прийняття рішень шляхом автоматизованої генерації агрономічних рекомендацій на основі формалізованих знань, бази даних і аналітичних алгоритмів.

Досягнення поставленої мети передбачає виконання таких **завдань**:

1. провести аналіз сучасного стану та потреб аграрного дорадництва;
2. дослідити предметну область фермерського господарства з точки зору інформаційного моделювання;

3. проаналізувати існуючі інформаційні системи дорадчого призначення;
4. сформулювати технічне завдання та визначити функціональні вимоги до майбутньої системи;
5. побудувати концептуальну архітектуру системи та функціональну модель у нотації IDEF0;
6. спроектувати базу даних та структуру інформаційних об'єктів;
7. реалізувати програмну частину системи засобами мови Python із використанням сучасних бібліотек;
8. провести тестування функціональності, зручності та надійності розробленої системи.

Об'єктом дослідження є система аграрного дорадництва як компонент інформаційної підтримки управлінських рішень у сільському господарстві.

Предметом дослідження виступає інформаційна система, що забезпечує автоматизоване формування аграрних порад на основі баз знань, вхідних даних та алгоритмів обробки.

Результати дослідження **мають прикладне значення** та можуть бути використані для створення практично орієнтованих інструментів підтримки фермерських господарств, що функціонують в умовах обмежених ресурсів. Розроблена система може бути адаптована до різних агрокліматичних зон та господарських моделей, що надає їй потенціал для масштабування.

Структура кваліфікаційної роботи включає вступ, три розділи, висновки, список використаних джерел та додатки. У першому розділі здійснено аналіз предметної області та постановку задачі. У другому розділі подано проектування архітектури, бази даних та функціональних компонентів системи. Третій розділ присвячено реалізації програмного забезпечення, перевірці працездатності та аналізу отриманих результатів.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Фермерське господарство в Україні становить основу малого аграрного виробництва та виступає важливою складовою в структурі продовольчої безпеки держави. Його діяльність організована на принципах приватної власності, самозайнятості та локалізованого управління. Типова структура фермерського господарства охоплює адміністративну функцію, виробничі підрозділи, забезпечення матеріально-технічними ресурсами, облік та контроль, а також інформаційно-консультаційний супровід [1]. У реальній практиці така структура часто є неформалізованою, що ускладнює ефективне прийняття управлінських рішень, оперативне реагування на зміну умов виробництва та впровадження інноваційних агротехнологій [2].

Особливої актуальності набуває питання інформаційної підтримки діяльності фермерів, що реалізується через дорадництво — механізм трансферу знань від експертного середовища до виробника. У рамках дорадчої діяльності циркулюють два основні типи інформаційних потоків: прямі (аналітичні звіти, методичні рекомендації, погодні моделі, ринкові прогнози) та зворотні (емпіричні дані з полів, запити, результати впровадження рекомендацій). Ці потоки взаємодіють у різних форматах — через консультаційні служби, вебресурси, тематичні додатки тощо [3].

На рис. 1.1 представлено узагальнену функціональну схему взаємодії інформаційних компонентів дорадницької системи.



Рисунок 1.1 – Структура функ

Вхідним елементом виступають агрономічні дані, що надходять до інформаційної системи з полів або зовнішніх сервісів (наприклад, метеостанцій, сенсорів, супутникових знімків). Центральний елемент — дорадницька інформаційна система — виконує роль посередника між базою знань та кінцевим користувачем. На основі внутрішньої експертної бази система формує персоналізовані рекомендації, адаптовані до конкретних умов господарства. Вихідний елемент — фермер — отримує агреговану, структуровану та практично застосовну інформацію.

Водночас існує низка системних проблем, які стримують ефективно впровадження дорадчих рішень. Серед основних бар'єрів слід виділити:

- фрагментарність існуючих джерел аграрної інформації;
- низький рівень цифровізації агросектору у сільській місцевості;
- відсутність адаптації порад до конкретних ґрунтово-кліматичних умов;
- обмежений зворотний зв'язок між дорадником і виробником.

Аналіз практик фермерських господарств показує, що понад 60 % аграріїв не мають постійного доступу до системного консультування, а 70 %

використовують неструктуровану інформацію з відкритих джерел, що знижує точність агротехнологічних рішень [4], [5].

Варто відзначити, що предметна область аграрного дорадництва 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

Станом на сьогодні в аграрному секторі впроваджено низку програмних рішень, орієнтованих на цифровізацію фермерського виробництва, моніторинг стану культур, управління ресурсами та прогнозування ризиків. Серед найбільш поширених систем варто виокремити AgroSmart, FieldView, Agrivi та Cropwise.

Система AgroSmart надає користувачам змогу отримувати агрометеорологічну аналітику, прогнози захворюваності культур, а також сповіщення про критичні мікрокліматичні умови. На рис. 1.2 зображено інтерфейс системи з візуалізацією карт ризику, побудованих на основі даних сенсорів вологості, температури та супутникового спостереження.

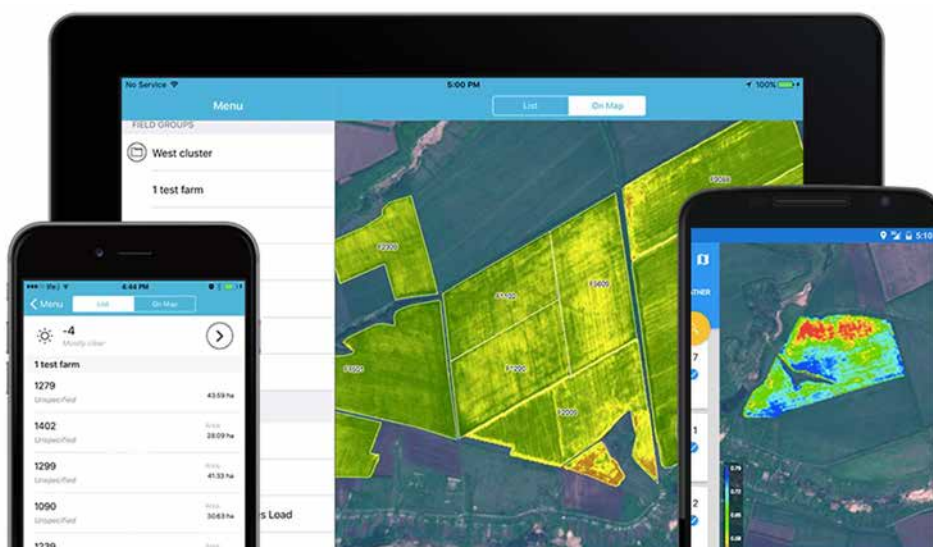


Рисунок 1.2 – Візуалізація мікрокліматичних загроз у системі AgroSmart

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

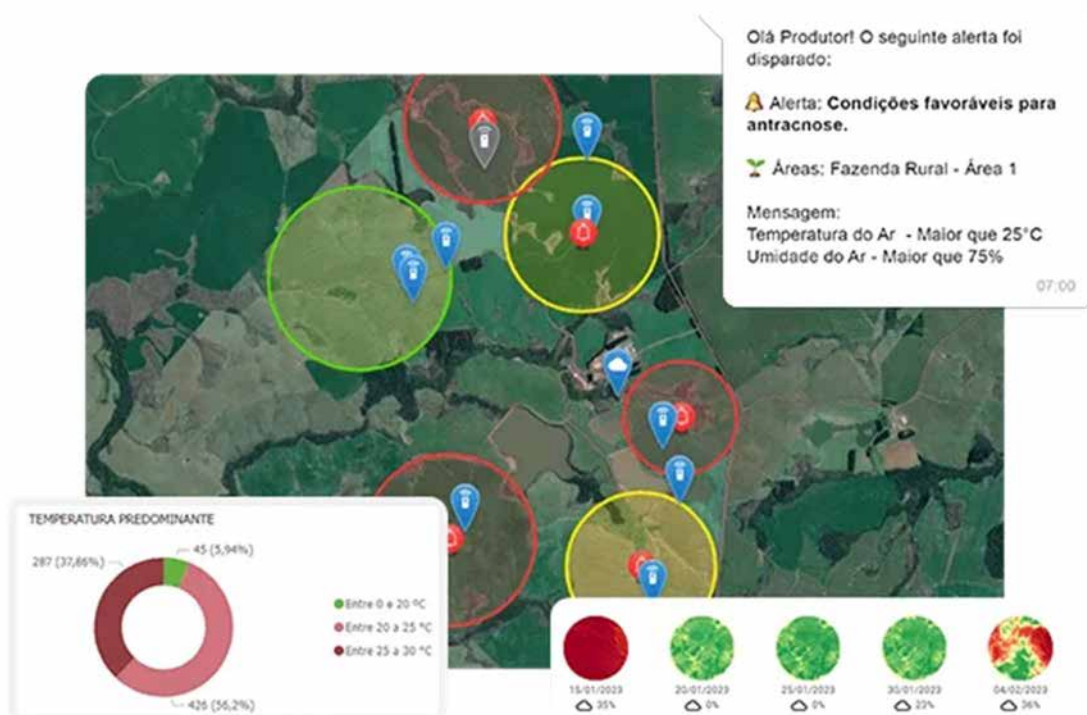


Рисунок 1.3 - Картографія зон врожайності у FieldView

Інтерфейс мобільного додатку забезпечує доступ до історії польових операцій, поточного стану посівів та погодних умов у реальному часі, що підвищує гнучкість планування агрономічних заходів [5].

Платформа Agrivi вирізняється комплексністю підходу до агровиробництва: підтримується управління всіма етапами — від посіву до збирання врожаю та післязбиральної логістики. На рис. 1.4 представлено інтерфейс платформи, що демонструє багатокomпонентну структуру: управління полями, дані метеорологічних станцій, статус агрономічних операцій.

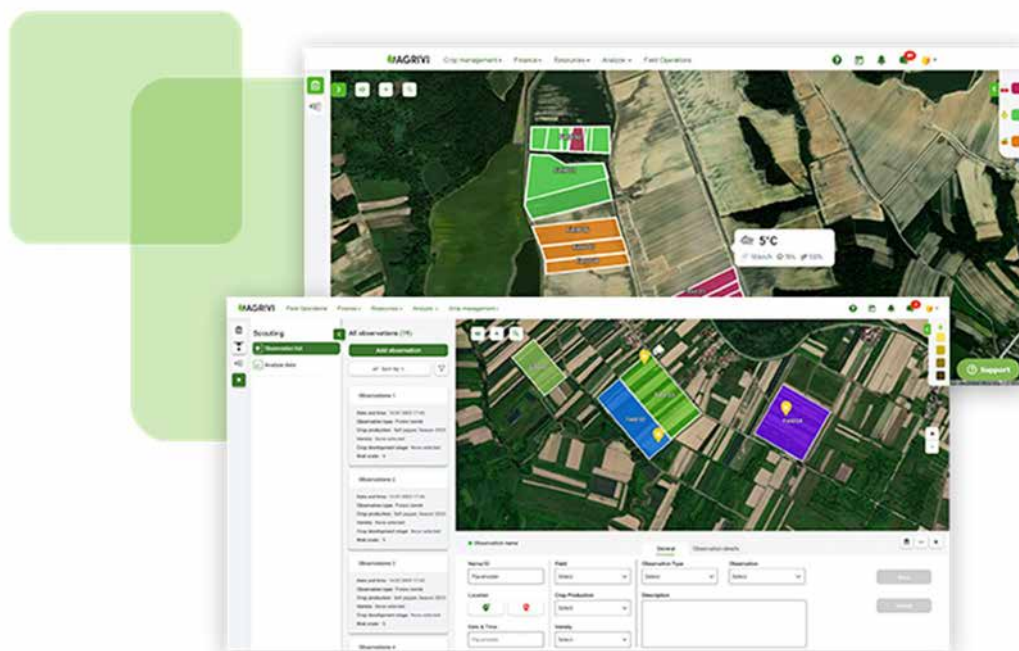


Рисунок 1.4 -Інтерфейс управління полями у платформі Agrivi

Присутня можливість призначення задач персоналу, автоматизоване формування звітів і підтримка багатокористувацького режиму [7].

Останній приклад — Cropwise Operations — характеризується простим та адаптивним інтерфейсом, оптимізованим під мобільні пристрої. Рис. 1.5 відображає карту зон продуктивності, побудовану за результатами аналізу біофізичних індексів.



Рисунок. 1.5 – Зони продуктивності поля у Cropwise Operations

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 фермерських господарств. Дані узагальнено в табл. 1.1.

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

Система	Функціональність	UI/UX	Локалізація	Доступність для малих господарств
AgroSmart	Метеодані, хвороби, аналіз ґрунту	Помірно складний інтерфейс	(переважно EN/PT)	(висока ціна)
Agrivi	Повний цикл виробництва, логістика	Інтуїтивно зрозумілий	(відсутня UA)	(комерційна підписка)
FieldView	Моніторинг техніки, карти врожайності	Простий та зручний	(лише EN)	(орієнтовано на великі господарства)
Cropwise Ops	Карты продуктивності, GPS інтеграція	Адаптовано до мобільних пристроїв	(є UA/RU)	(є безкоштовний функціонал)

Аналіз свідчить, що більшість платформ демонструють високий рівень функціональності, однак потребують значних фінансових ресурсів для повноцінного використання або орієнтовані на великотоварне виробництво [4], [7], [9]. Водночас інтерфейси користувача часто перевантажені або не локалізовані, що знижує ефективність взаємодії для україномовного фермерського сегменту [2], [5].

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 англomовної підготовки [2], [6].
2. Недоступність для малого бізнесу через високу ціну ліцензії, підписки або необхідність у спеціалізованому обладнанні [1], [4].
3. Обмежена персоналізація порад, коли системи не враховують специфіку мікроклімату, ґрунтів або виробничих потужностей конкретного господарства [3], [7].
4. Закритість архітектури та неможливість адаптації системи до локальних умов без втручання розробника [8].

Актуальним залишається завдання розробки інформаційної системи аграрного дорадництва, 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 [5], [10].

1.3 Визначення вимог до інструментальних засобів та розробка технічного завдання

Проектування інформаційної системи дорадницької підтримки фермерських господарств 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, супутникові знімки, польові

сенсори), генерація рекомендацій на основі правил та моделей, формування короткострокових прогнозів, передача повідомлень користувачу та збереження історичних даних в структурованому форматі [3], [5], [6].

Узагальнення функціональних вимог подано в табл. 1.2.

Таблиця 1.2 – Функціональні вимоги до системи аграрного дорадництва

Компонент	Опис функціональності
Інформаційний блок	Отримання даних з API, супутникових знімків, ручне введення польової інформації
Генерація порад	Автоматизоване формування рекомендацій з бази знань залежно від вхідних параметрів
Прогнозування	Побудова короткострокових погодніх і агротехнологічних прогнозів
Повідомлення	Сповіщення про ризики, рекомендації, зміни погодніх умов через інтерфейс
Облік даних	Збереження історії запитів, результатів, змін у параметрах ділянок

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 [1], [4], [7].

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 через браузер без потреби у встановленні ПЗ, адаптація до слабких мереж
Надійність	Автоматичне збереження даних, обробка виняткових ситуацій
Локалізація	Повна 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 [6], [7].

Вхідними параметрами є агрономічні та технологічні характеристики, що характеризують конкретне господарство. Зокрема, передбачено інтеграцію таких типів даних, як геопросторові координати ділянки, тип ґрунту, площа, вирощувана культура, погодні параметри (температура, вологість, атмосферний тиск, рівень опадів), а також історичні агротехнічні події (дата посіву, дата обробки, внесення добрив). З технічної точки зору, ці дані можуть бути подані у форматі JSON або CSV при імпорті через інтерфейс, або отримані автоматично через REST-з'єднання з зовнішніми сервісами (метеостанції, супутники, сенсори) [3], [5].

На основі зібраних даних система має генерувати вихідну інформацію, що включає персоналізовані рекомендації (наприклад, час внесення добрив або захисту від шкідників), погодні попередження, а також візуалізацію даних у вигляді графіків, теплових карт і діаграм. Рекомендації повинні формуватись із використанням бази знань, що реалізує продукційні правила, а також із залученням емпіричних алгоритмів аналізу [1], [4], [9].

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. 1.6 продемонстровано взаємозв'язки між цими елементами, що дозволяє краще уявити узгодженість архітектурних вимог з функціональним призначенням майбутньої системи.

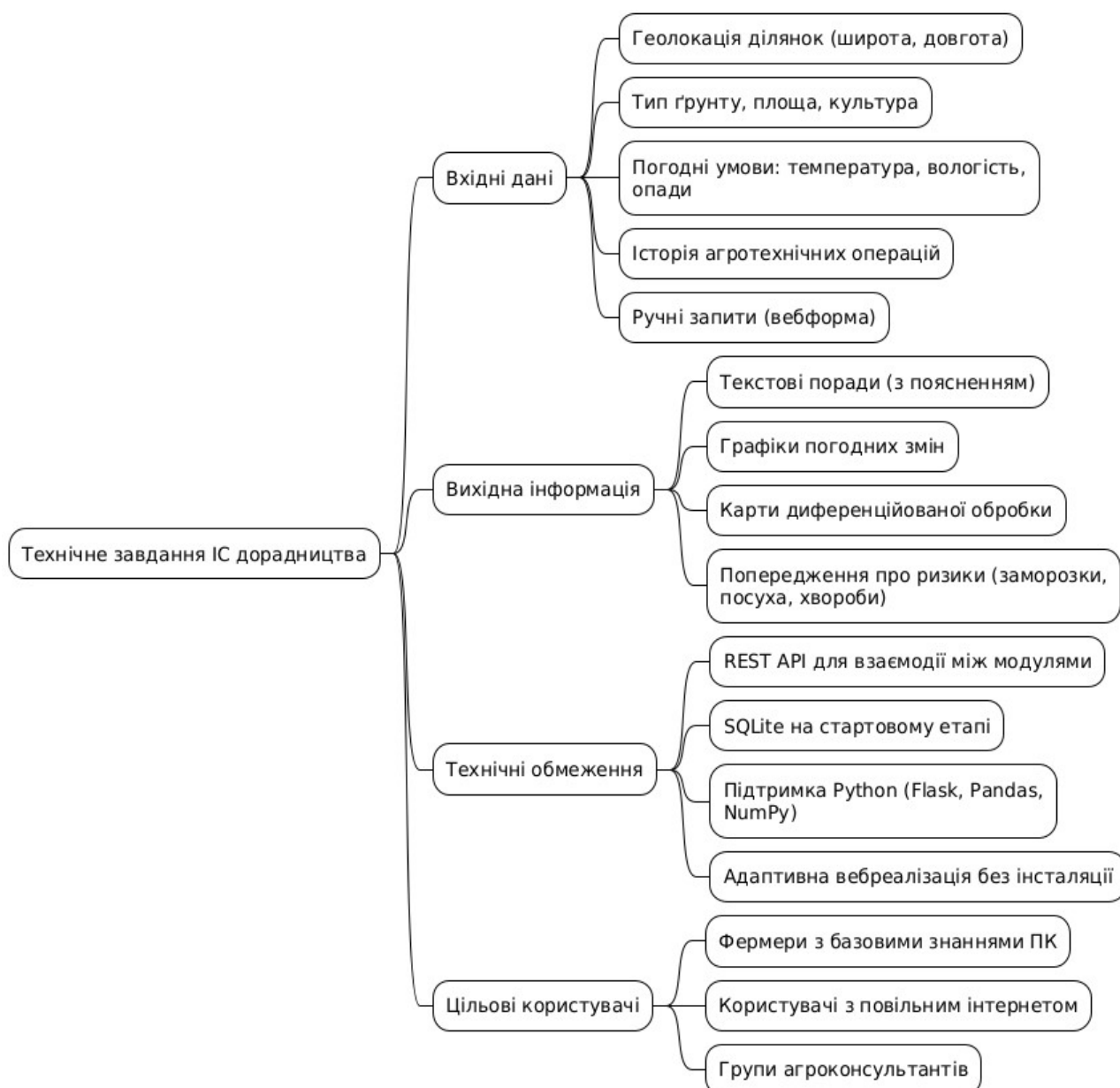


Рисунок 1.6 – Інтелект-карта технічного завдання системи аграрного дорадництва

Розробка системи повинна враховувати ряд обмежень. Зокрема, реалізація має ґрунтуватись на технологіях із відкритим кодом, використовувати SQLite як початкову систему керування базами даних із можливістю подальшого масштабування до MySQL або PostgreSQL. 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 (наприклад, Flask, Pandas,

Matplotlib) повинні забезпечувати обробку, агрегацію, візуалізацію та маршрутизацію даних у реальному часі [21], [23], [30].

Цільовими користувачами системи визначено малі та середні фермерські господарства, зокрема ті, що мають обмежений технічний ресурс і базовий рівень цифрової грамотності. Тому особлива увага приділяється зручності інтерфейсу, мінімізації кількості дій для отримання поради, а також автоматизації процесів, що не потребують втручання користувача [2], [8].

1.4 Висновок до першого розділу

У результаті виконаного дослідження сформовано цілісне уявлення про предметну область аграрного дорадництва в умовах цифрової трансформації агросектору. Проаналізовано структурні та функціональні особливості фермерських господарств, 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 [1], [4], [6].

Оцінка сучасних IT-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 [5], [7], [9].

На основі виявлених обмежень сформульовано функціональні та нефункціональні вимоги до майбутньої інформаційної системи аграрного дорадництва. 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). Технічне завдання передбачає реалізацію веборієнтованої, адаптивної системи з відкритою архітектурою, що відповідає вимогам масштабованості, доступності та простоти використання для фермерів із базовим рівнем цифрової грамотності [2], [8], [10].

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 ДОРАДНИЦЬКОЇ 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 дорадницької інформаційної системи фермерського господарства 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.

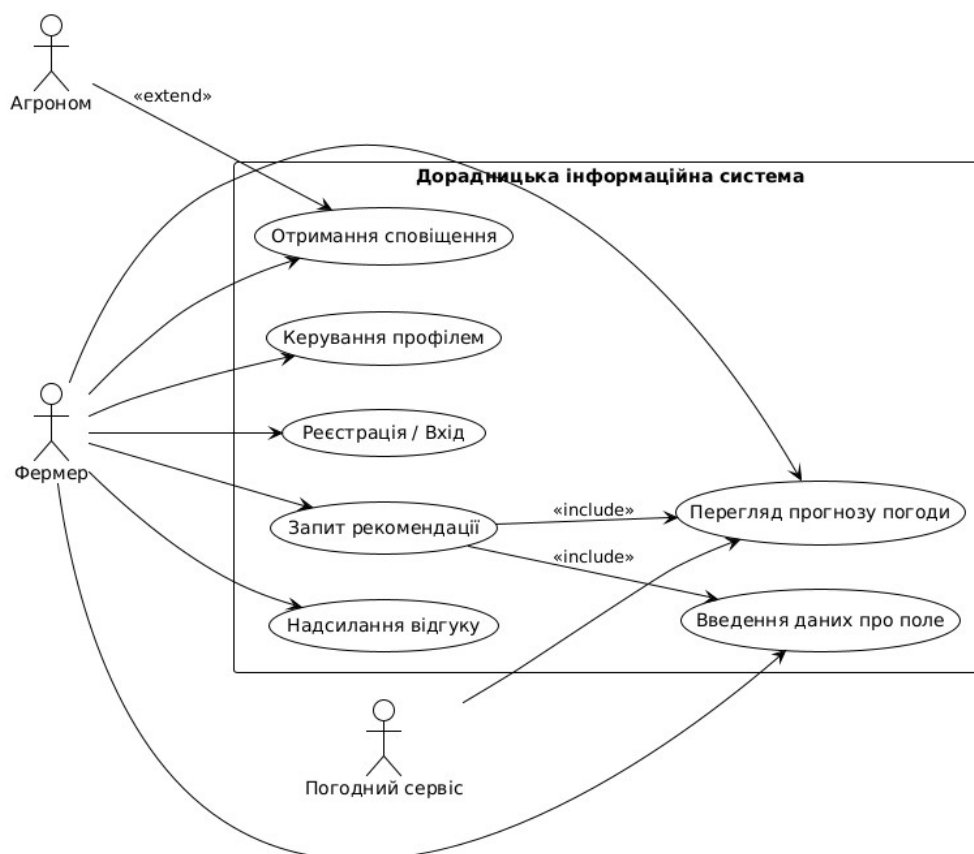


Рисунок 2.1– Діаграма прецедентів дорадницької інформаційної системи

Основним користувачем виступає фермер, що має доступ до базового функціоналу: реєстрація або вхід, введення агрономічних даних про поле, запит агрономічних рекомендацій, перегляд метеопрогнозів, керування особистим профілем, а також надсилання зворотного зв'язку після отримання поради. Крім того, до системи підключається погодні сервіс — зовнішній актор, що забезпечує надходження актуальних кліматичних даних через API, а також агроном — експерт, який має доступ до отримання системних сповіщень та моніторингу дій користувачів. Зв'язки між прецедентами позначені відношеннями <<include>> для обов'язкових дій та <<extend>> для додаткової логіки. Наприклад, прецедент “Запит рекомендації” включає попередні дії “Введення даних про поле” та “Перегляд прогнозу погоди”, оскільки ці кроки є критично необхідними для формування достовірної поради.

У рамках динамічного аналізу предметної області побудовано діаграму послідовності (рис. 2.2), яка описує обмін повідомленнями між логічними компонентами системи в сценарії генерації рекомендації.



Рисунок 2.2 – Діаграма послідовності сценарію генерації рекомендації

Процес починається з того, що фермер вводить дані про стан поля через користувацький інтерфейс. Далі ці дані надходять до ядра системи, яке ініціює звернення до зовнішнього погодного API з метою отримання метеоумов.

Після надходження зовнішньої інформації система звертається до внутрішньої бази знань, у якій зберігаються формалізовані агрономічні правила та алгоритми обробки. На основі поєднання вхідних даних і наявних знань формується порада, що передається користувачу через інтерфейс. Після ознайомлення з результатом користувач має можливість залишити відгук, який зберігається в системі для подальшого аналізу ефективності рекомендаційного модуля.

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

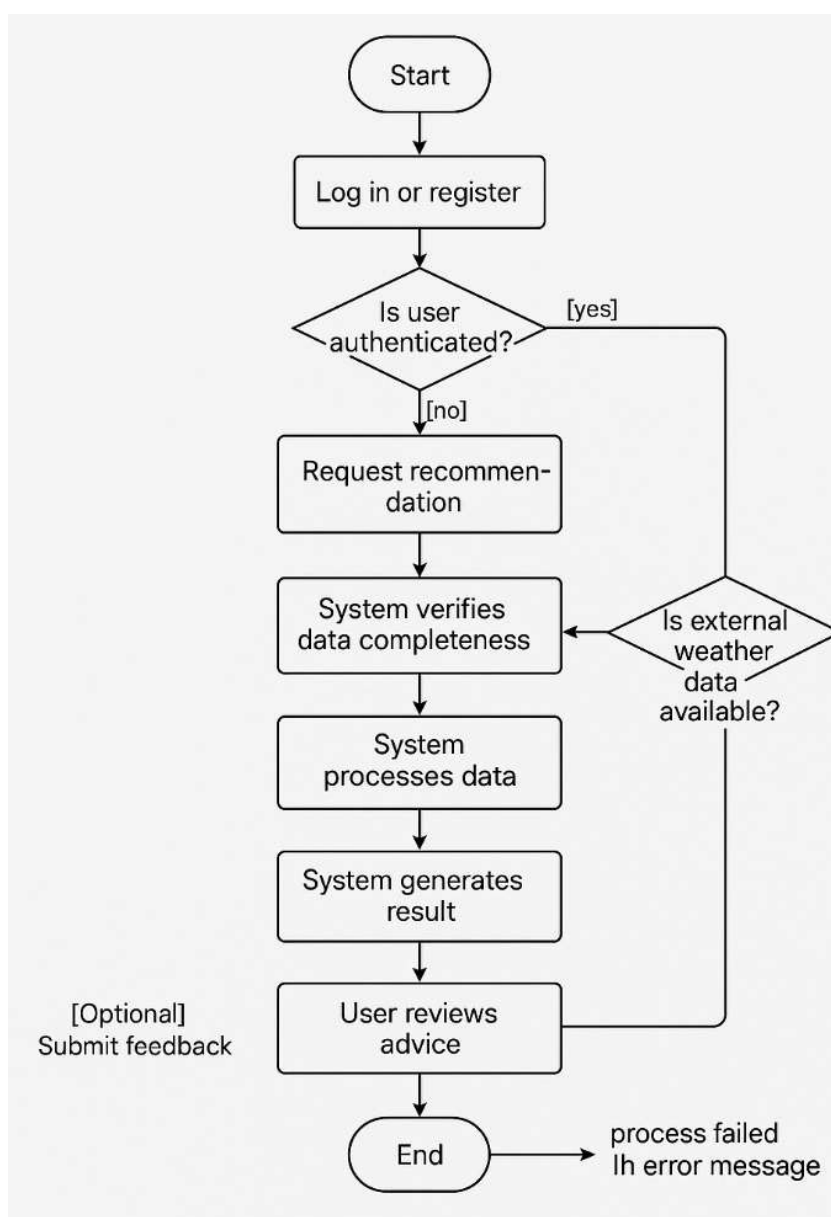


Рисунок 2.3– Діаграма активності процесу запиту агрономічної поради

Ця діаграма моделює алгоритм дій системи у випадку типового сценарію: від моменту автентифікації користувача до виведення результату. Після успішного входу фермер ініціює запит рекомендації, вводить або оновлює дані про поле, після чого система перевіряє повноту введених значень. Якщо всі обов'язкові параметри заповнені, система звертається до погодного сервісу. У разі недоступності зовнішнього джерела процес завершується повідомленням про помилку. Якщо погодні дані отримано, запускається обробка інформації, згенерований результат відображається користувачу. Додатково реалізовано опціональний блок надсилання відгуку. Завершенням процесу є повернення до головного меню або вихід із системи.

Застосування UML-нотацій у моделюванні предметної області дозволило формалізувати функціональну модель системи, окреслити структуру взаємодій між основними компонентами, встановити сценарії поведінки користувача та визначити потенційні точки відмови. Побудовані діаграми слугують фундаментом для подальшого проєктування архітектури системи, структури бази даних та реалізації логіки окремих програмних модулів. Такий підхід забезпечує логічну цілісність і узгодженість між етапами життєвого циклу створення інформаційної системи.

2.2 Архітектура програмного забезпечення

Розробка програмного забезпечення дорадницької інформаційної системи 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 рис. 2.4 представлено архітектуру програмного забезпечення, що демонструє взаємозв'язки між основними логічними блоками системи.

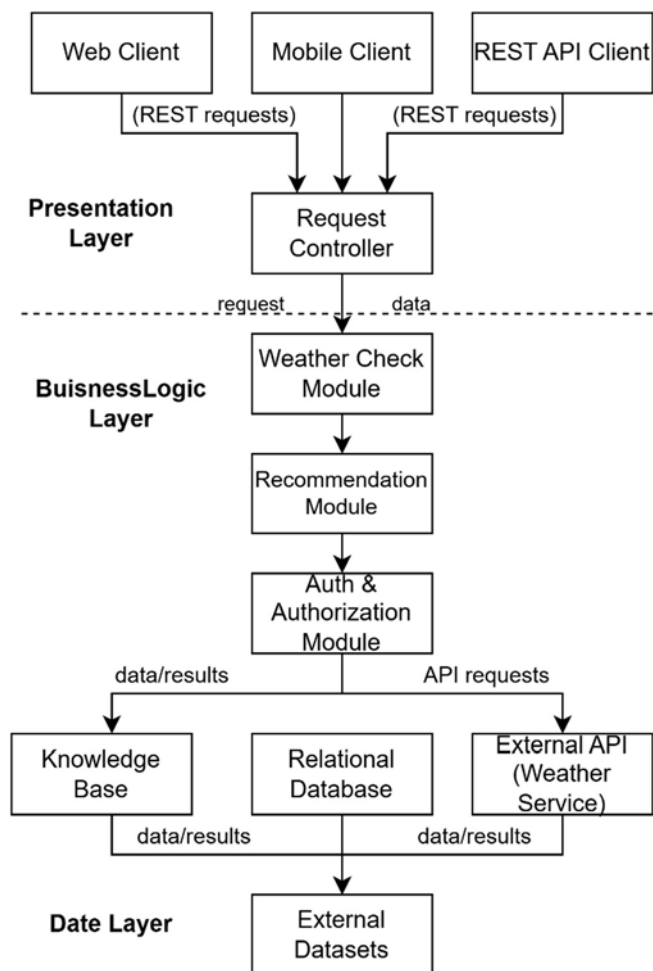


Рисунок 2.4— Багаторівнева архітектура дорадницької інформаційної системи

Верхній рівень — Presentation — включає інтерфейсні компоненти, з якими взаємодіє користувач: вебінтерфейс, мобільний застосунок та REST API клієнт. Ці компоненти відповідають за введення агрономічних даних, запити рекомендацій та перегляд результатів.

Середній рівень — Business Logic — реалізує обробку запитів, логіку взаємодії з базою знань і зовнішніми сервісами, авторизацію та зворотний зв'язок. До його складу входять: контролер користувацьких запитів, модуль перевірки погодних умов, генератор агропорад, модуль автентифікації, а також менеджер відгуків. Кожен із цих компонентів виконує строго визначену функцію, що підвищує підтримуваність та стабільність системи при масштабуванні.

Нижній рівень — Data Layer — відповідає за зберігання і доступ до агрономічної інформації. Він включає реляційну базу даних для зберігання даних користувача, історії запитів та агрономічних характеристик, базу знань з правилами формування рекомендацій, а також доступ до зовнішніх API, таких як погодні сервіси.

Загалом архітектура системи забезпечує централізовану обробку агрономічних даних, автоматизовану генерацію порад на основі формалізованих знань і контекстних параметрів, а також підтримку адаптивної взаємодії з користувачем. Кожен логічний компонент описано у табл. 2.1.

Таблиця 2.1– Основні компоненти архітектури системи

№	Компонент	Призначення
1	Web Interface	Вебінтерфейс для введення даних і перегляду результатів
2	Mobile Application	Мобільний застосунок для роботи з системою в польових умовах
3	REST API Client	Канал взаємодії з системою через зовнішні сервіси або скрипти
4	User Request Controller	Обробка запитів користувача та ініціація внутрішніх процесів
5	Weather Check Module	Звернення до зовнішнього API та аналіз погодних умов
6	Agro-Advice Generation Module	Формування агропорад на основі бази знань та вхідних параметрів
7	Authorization and Authentication Service	Захист доступу до системи, управління сесіями користувачів
8	Feedback Manager	Збір, обробка та зберігання відгуків користувачів
9	Knowledge Base	Формалізовані агрономічні правила, що лежать в основі алгоритмів системи
10	Relational Database	Сховище користувацьких, історичних та операційних даних
11	External API	Інтеграція з зовнішніми сервісами (наприклад, погодними)

Запропонована архітектура є придатною як для централізованого, так і для розподіленого розгортання. Вона забезпечує чітке розмежування обов'язків, масштабованість логіки, незалежність від конкретного інтерфейсу, а також

сумісність з існуючими погодними платформами. Такий підхід дозволяє гнучко адаптувати систему до змін вимог, збільшення кількості користувачів чи розширення функціоналу.

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

Функціональні модулі є ядром дорадницької інформаційної системи, 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 моделі IDEF0 (функціональна декомпозиція), BPMN (бізнес-процес) та IDEF5 (онтологічна схема).

На рис. 2.5 подано функціонально-структурну модель у нотації IDEF0, яка відображає верхньорівневе представлення функції “Формування агрономічної поради” з подальшою декомпозицією.

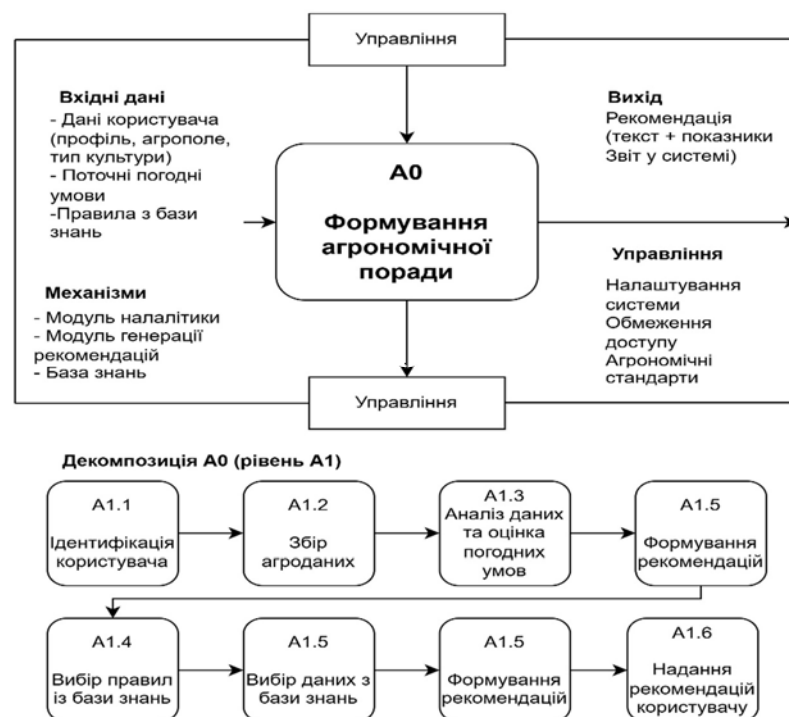


Рисунок 2.5– IDEF0-модель формування агрономічної поради (рівні А0–А1)

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.

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 знань, виконує агроаналіз, а потім будує рекомендацію, яка надається користувачу.

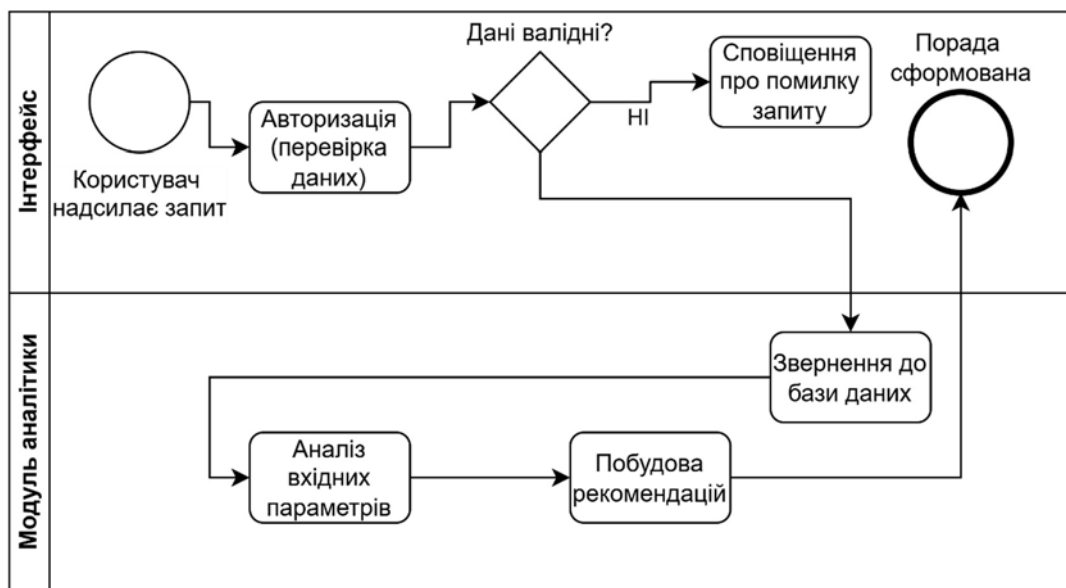


Рисунок 2.6– BPMN-діаграма бізнес-процесу генерації агрономічної поради

Для опису внутрішньої структури понять і зв'язків між об'єктами побудовано онтологічну модель у нотації IDEF5 (рис. 2.7). Модель ілюструє відношення між класами “Користувач”, “Запит”, “Модуль аналітики”, “Правило” та “База знань”. Зокрема, користувач генерує запит, який активує модуль аналітики, що звертається до бази знань.

Правила мають тип, структуру, мету і час дії, а також пов'язані з результатами — рекомендаціями, що повертаються користувачу.

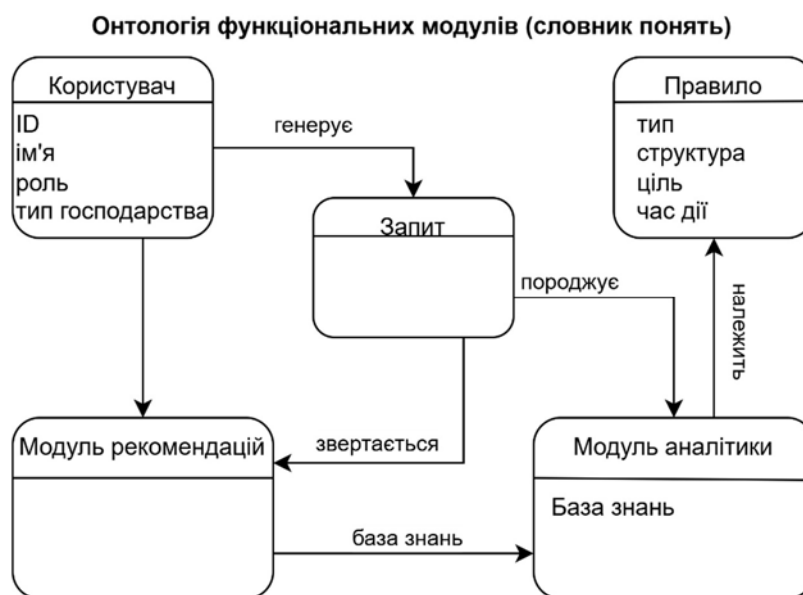


Рисунок 2.7– Онтологічна модель функціональних модулів у нотації IDEF5

Характеристики основних функціональних модулів наведено у табл. 2.2.

Таблиця 2.2– Характеристика основних функціональних модулів системи

№	Назва модуля	Призначення
1	Модуль аналітики	Обробляє вхідні дані, визначає релевантні параметри, активує вибір правил
2	Модуль рекомендацій	Формує кінцеву пораду на основі правил, погодних умов і профілю користувача
3	База знань	Містить формалізовані агрономічні правила, оновлювані адміністраторами
4	Модуль користувачів	Забезпечує автентифікацію, зберігає профілі, історію запитів та фідбек

Узагальнюючи, представлена система функціональних модулів реалізує повний цикл обробки запиту – від збору агрономічної інформації до доставки персоналізованої рекомендації та фіксації зворотного зв'язку. 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 дорадницької інформаційної системи 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, спрощує розгортання на стороні користувача та дає змогу підтримувати офлайн-режим.

На рис. 2.8 представлено компонентну архітектуру системи, яка побудована за принципами слабого зв'язку між модулями.

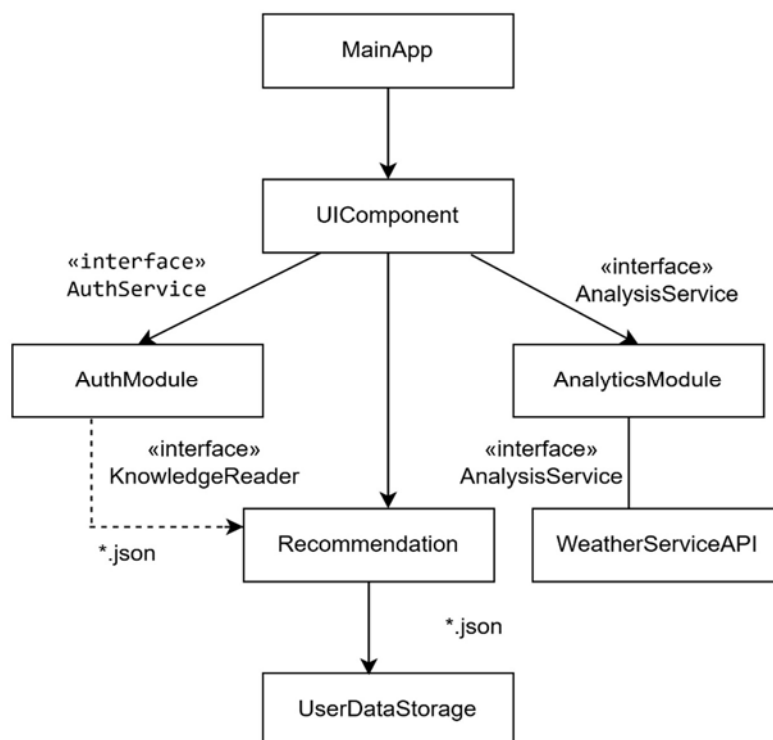


Рисунок 2.8– Компонентна архітектура дорадницької інформаційної системи на Java

Центральний компонент `UIComponent` взаємодіє з модулями авторизації, аналітики та генерації поради через відповідні інтерфейси. Компонент `Recommendation` об'єднує логіку формування агропорад на основі даних з `AnalyticsModule`, `WeatherServiceAPI` та `KnowledgeBase`, а результати записуються у `UserDataStorage` у форматі `.json`. Головний клас `MainApp` відповідає за ініціалізацію всієї системи.

Характеристики вибраних інструментальних засобів зведено у табл. 2.3.

Таблиця 2.3– Основні засоби реалізації системи

№	Компонент	Засіб реалізації	Призначення
1	Мова програмування	Java 17+	Базова логіка, обробка подій, файловий доступ
2	Графічний інтерфейс	Java Swing	Візуальні форми: введення, результат, зворотний зв'язок
3	Формат зберігання	JSON (org.json, Gson)	Дані користувача, рекомендації, історія, правила агроаналізу
4	Локальна база знань	JSON-файл (rules.json)	Статичні агрономічні правила для системи експертного типу

Продовження таблиці 2.3

5	Аналіз погодних умов	Емулятор WeatherServiceAPI	Псевдо-API для оцінки погодних параметрів при розрахунках
6	Авторизація користувачів	Локальний модуль AuthModule	Обробка даних входу, контроль сесії користувача

Обрані засоби дозволяють реалізувати гнучку, адаптовану до користувача систему, з мінімальними інфраструктурними вимогами. Такий підхід є доцільним у контексті фермерських господарств, 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. 2.9 представлено ER-діаграму, яка описує основні сутності, їх атрибути та типи зв'язків між ними.

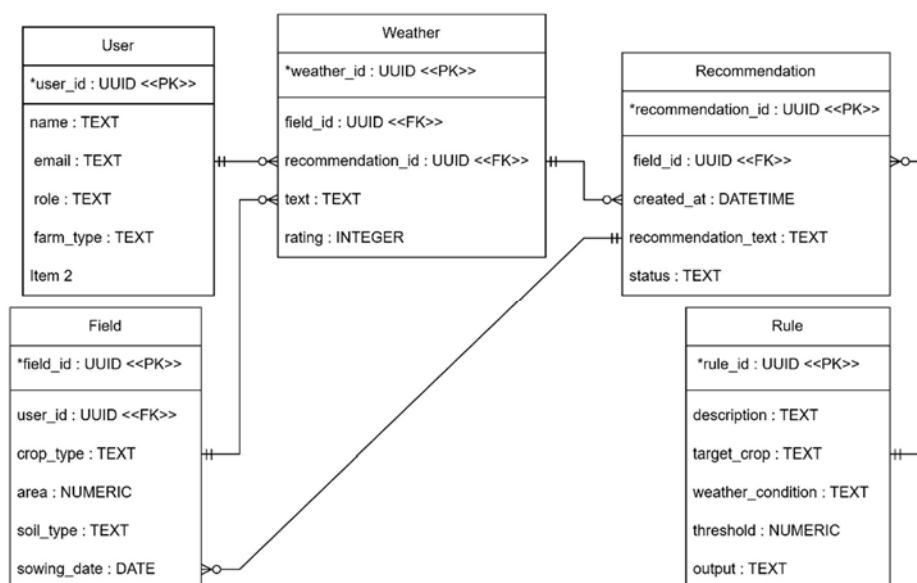


Рисунок 2.9– ER-діаграма бази даних дорадницької інформаційної системи

Центральною сутністю є User, яка представляє користувача системи — фермера або агронома. З кожним користувачем може бути пов'язано декілька об'єктів Field (поля або ділянки), для яких фіксуються Weather-умови та формуються Recommendation-поради на основі правил з Rule.

Сутність Recommendation зберігає сформовану рекомендацію, пов'язану з конкретним полем. Вона також містить стан реалізації поради (status) — застосована чи ні. Правила в Rule пов'язуються з рекомендаціями логічно через параметри target_crop, weather_condition та threshold. Дані про погоду зберігаються у сутності Weather та включають температуру, вологість, опади тощо, пов'язані із полем і конкретною рекомендацією.

Зведені характеристики основних таблиць системи наведено в табл. 2.4.

Таблиця 2.4– Логічна структура основних таблиць бази даних

Таблиця	Поле	Тип даних	Призначення
User	user_id (PK)	UUID / TEXT	Унікальний ідентифікатор користувача
	name	TEXT	Ім'я користувача
	email	TEXT	Адреса електронної пошти
	role	TEXT	Роль (farmer, agronom)
	farm_type	TEXT	Тип господарства
Field	field_id (PK)	UUID / TEXT	Ідентифікатор поля
	user_id (FK)	UUID / TEXT	Зовнішній ключ на користувача
	crop_type	TEXT	Тип культури
	area	NUMERIC	Площа поля
	soil_type	TEXT	Тип ґрунту
Recommendation	sowing_date	DATE	Дата висіву
	recommendation_id (PK)	UUID / TEXT	Унікальний ідентифікатор поради
	field_id (FK)	UUID / TEXT	Посилання на поле
	created_at	DATETIME	Дата створення поради
	recommendation_text	TEXT	Вміст поради
Weather	status	TEXT	Стан: applied / not applied
	weather_id (PK)	UUID / TEXT	Ідентифікатор погодного запису

Продовження таблиці 2.4

	field_id (FK)	UUID / TEXT	Посилання на поле
	recommendation_id (FK)	UUID / TEXT	Посилання на пораду
	text	TEXT	Узагальнений опис умов
	rating	INTEGER	Оцінка погодного впливу
Rule	rule_id (PK)	UUID / TEXT	Унікальний ідентифікатор правила
	description	TEXT	Опис правила
	target_crop	TEXT	Цільова культура
	weather_condition	TEXT	Умова (напр. >25°C, вологість>75%)
	threshold	NUMERIC	Поріг активації правила
	output	TEXT	Результат застосування правила

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.

2.6 Опис інтерфейсу користувача

Інтерфейс користувача (UI) дорадницької інформаційної системи 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.

На рис. 2.10 представлено макет екрана інтерфейсу, що демонструє приклад рекомендаційного звіту з аналізом метрик поля. Ліва частина містить таблицю з базовими показниками (яскравість, контраст, ентропія), побудовану на основі обробленого зображення. Праворуч розміщено гістограму яскравості як візуальний супровід для підтвердження структури даних. У центрі виводиться узагальнений HTML-звіт із загальним результатом — "Healthy" — який доповнюється текстовою інтерпретацією. Нижче наведено формульні

характеристики та індекси, які дозволяють агроному провести самостійну оцінку.

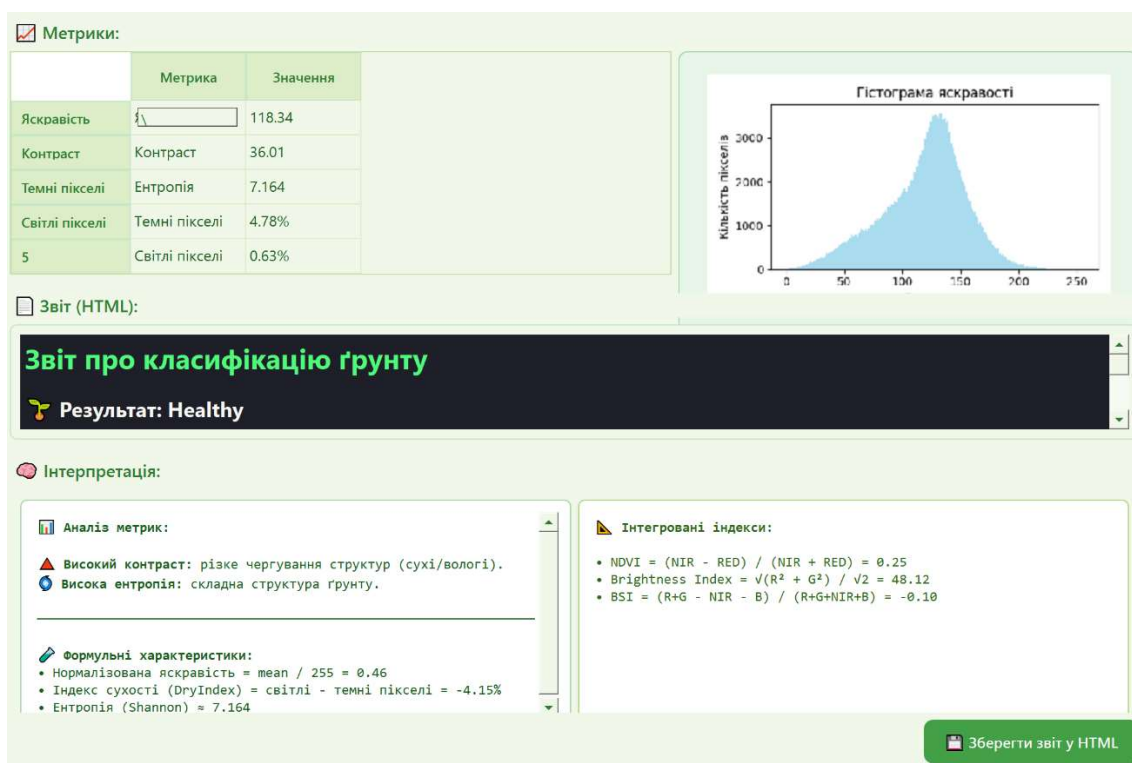


Рисунок 2.10– Інтерфейс виводу результатів агроаналізу з аналітичним та HTML-звітом

Інтерфейс реалізовано у стилі сільськогосподарської тематики з використанням зелених та м'яких пастельних відтінків, що не перевантажують візуальне сприйняття. Табл. 2.5 узагальнює основні функціональні елементи інтерфейсу та їх призначення.

Таблиця 2.5– Функціональні елементи інтерфейсу користувача

№	Елемент	Призначення
1	Таблиця метрик	Вивід значень яскравості, контрасту, ентропії, кількості темних/світлих пікселів
2	Гістограма яскравості	Графічне представлення розподілу яскравості пікселів у полі
3	HTML-звіт	Автоматично згенерований текстовий висновок за результатами аналізу
4	Блок інтерпретації	Пояснення аналітичних висновків (контраст, ентропія, сухість)
5	Формули обчислення	Виведення індексів: NDVI, BSI, Brightness Index, DryIndex

6	Кнопка збереження звіту	Збереження звіту у HTML-форматі на локальний диск користувача
---	-------------------------	---

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.

2.7 Висновок до другого розділу

У результаті проведеного проектування сформовано цілісну концептуальну модель дорадницької інформаційної системи для фермерських господарств, що враховує особливості предметної області, вимоги користувачів та технічні обмеження середовища застосування.

Виконано моделювання предметної області із застосуванням UML-нотацій (діаграми прецедентів, послідовності, активностей), що дозволило формалізувати сценарії взаємодії користувачів із системою, визначити критичні процеси генерації агрономічних рекомендацій та побудувати поведінкові моделі. Діаграми відображають логіку обробки агрономічних даних, інтеграцію з погодними сервісами, а також маршрути обробки запитів та зворотного зв'язку.

Запропоновано багаторівневу архітектуру програмного забезпечення, побудовану за принципами модульності та розділення відповідальностей. Система структурована на презентаційний рівень (вебінтерфейс, мобільний застосунок, REST API), рівень бізнес-логіки (модулі аналітики, генерації порад, автентифікації, зворотного зв'язку) та рівень доступу до даних (база знань,

агродані, зовнішні API). Такий підхід забезпечує гнучкість, масштабованість та незалежність від конкретного середовища розгортання.

Описано функціональні модулі системи за допомогою IDEF0, BPMN та IDEF5 моделей. Побудовані схеми демонструють логічну цілісність системи, узгодженість компонентів, послідовність обробки агрономічних запитів та формування вихідних рекомендацій. Створено онтологічну модель, що описує зв'язки між сутностямищо закладає підґрунтя для семантичної обробки даних.

Обґрунтовано вибір інструментальних засобів реалізації: Java як мова програмування, Java Swing для створення графічного інтерфейсу, формат JSON для зберігання агрономічної інформації та локальної бази знань. Така конфігурація забезпечує автономність роботи системи, мінімальні вимоги до інфраструктури, підтримку офлайн-режиму та зручність використання в умовах нестабільного інтернет-з'єднання.

Розроблено логічну модель бази даних у вигляді ER-діаграми, що охоплює сутності користувачів, полів, погодних умов, рекомендацій та правил. Побудована структура забезпечує узгодженість, розширюваність і підтримку реляційних або документно-орієнтованих моделей зберігання.

Описано інтерфейс користувача, орієнтований на доступність і простоту використання для осіб без спеціальної технічної підготовки. Реалізовано візуалізацію агропараметрів, динамічне оновлення рекомендацій та інтерактивну інтерпретацію результатів у HTML-форматі.

Отримані результати формують повноцінну основу для реалізації програмного прототипу системи.

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 (див. попередній розділ). Кожна сутність (користувач, поле, рекомендація, погода) серіалізується в окремий масив об'єктів у відповідному файлі (users.json, fields.json, recommendations.json, weather.json, rules.json).

На прикладі модуля авторизації продемонстровано основи реалізації обробки користувацьких даних, наведено в лістингу 3.1:

Лістинг 3.1

```
public class AuthModule {
    private final String USERS_FILE = "data/users.json";
    private List<User> users;

    public AuthModule() {
        loadUsers();
    }
    private void loadUsers() {
        try (Reader reader = new FileReader(USERS_FILE)) {
            Gson gson = new Gson();
            Type userListType = new TypeToken<List<User>>().getType();
            users = gson.fromJson(reader, userListType);
        } catch (IOException e) {
            users = new ArrayList<>();
        }
    }
    public boolean authenticate(String email, String password) {
        return users.stream()
            .anyMatch(user -> user.getEmail().equals(email) &&
                user.getPassword().equals(password));
    }
}
```

Користувачі зберігаються у файлі `users.json` у вигляді структурованого масиву, відображено в лістингу 3.2:

Лістинг 3.2

```
[
  {
    "user_id": "u001",
    "name": "Іван Петренко",
    "email": "ivan@example.com",
    "password": "1234",
    "role": "farmer",
    "farm_type": "Змішане"
  }
]
```

Подібним чином реалізовано зчитування агропараметрів із файлу `fields.json`. Модуль зберігання полів реалізовано через об'єктну модель, наведено в лістингу 3.3:

Лістинг 3.3

```
public class FieldModule {
    private List<Field> fields;
    private final String FIELD_FILE = "data/fields.json";

    public void saveField(Field field) {
        fields.add(field);
        try (Writer writer = new FileWriter(FIELD_FILE)) {
            new GsonBuilder().setPrettyPrinting().create().toJson(fields,
writer);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Приклад JSON-запису про агрополе наведено в лістингу 3.4:

Лістинг 3.4

```
{
  "field_id": "f001",
  "user_id": "u001",
  "crop_type": "Кукурудза",
  "area": 12.5,
  "soil_type": "Супіщаний",
  "sowing_date": "2024-04-10"
}
```

У модулі рекомендацій реалізовано алгоритм вибору правила з бази знань та формування поради на основі погодних умов і типу культури представлено в лістингу 3.5:

Лістинг 3.5

```
public class RecommendationEngine {
    private List<Rule> rules;

    public Recommendation generate(Field field, Weather weather) {
        for (Rule rule : rules) {
            if (rule.getTargetCrop().equals(field.getCropType())
                && weather.getTemperature() >= rule.getThreshold()) {
                return new Recommendation(field.getFieldId(),
                    rule.getOutput(), LocalDateTime.now(), "not applied");
            }
        }
        return new Recommendation(field.getFieldId(),
            "Недостатньо даних для рекомендації", LocalDateTime.now(), "not
applied");
    }
}
```

Приклад JSON-запису поради, що представлено в лістингу 3.6:

Лістинг 3.6

```
{
    "recommendation_id": "r001",
    "field_id": "f001",
    "created_at": "2025-05-01T10:12:00",
    "recommendation_text": "Застосувати боронування в найближчі 2 дні",
    "status": "not applied"
}
```

Аналогічно реалізовано модуль погоди WeatherServiceEmulator, що зчитує умовні параметри для конкретного поля, JSON-запис якого представлений в лістингу 3.7:

Лістинг 3.7

```
{
    "weather_id": "w001",
    "field_id": "f001",
    "recommendation_id": "r001",
    "text": "Температура: 27°C, Вологість: 70%",
    "rating": 4
}
```

Зведені функції реалізованих модулів наведено у табл. 3.1.

Таблиця 3.1– Основні реалізовані модулі системи

№	Модуль	Призначення
1	AuthModule	Завантаження користувачів, перевірка автентифікації
2	FieldModule	Збереження та завантаження даних про поля користувача
3	RecommendationEngine	Вибір правил із бази знань, формування поради на основі вхідних параметрів
4	WeatherServiceAPI	Зчитування або емуляція погодних умов для конкретної ділянки
5	JsonStorageService	Універсальний інтерфейс для читання/запису в JSON-файли

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

3.2 Розробка алгоритмів обробки даних та генерації порад

Основою функціональності дорадницької інформаційної системи є алгоритми, які забезпечують аналіз аграрних параметрів, погодних умов та визначення релевантних рекомендацій для конкретного поля. Розроблені алгоритми реалізують обробку даних, отриманих від користувача (тип культури, площа, тип ґрунту, дата посіву), і даних із симульованого погодного джерела, що зберігаються у форматі JSON. Виходом є персоналізована текстова порада, сформована на основі агрономічних правил.

Розробка алгоритму формування поради базується на моделі зіставлення агропараметрів із правилами з бази знань. Умови кожного правила задаються у вигляді: культура + погодна умова + поріг, а вихід — текст поради. Для реалізації вибору актуального правила було розроблено відповідний модуль фільтрації. Блок-схема алгоритму обробки даних представлений на рис.3.8

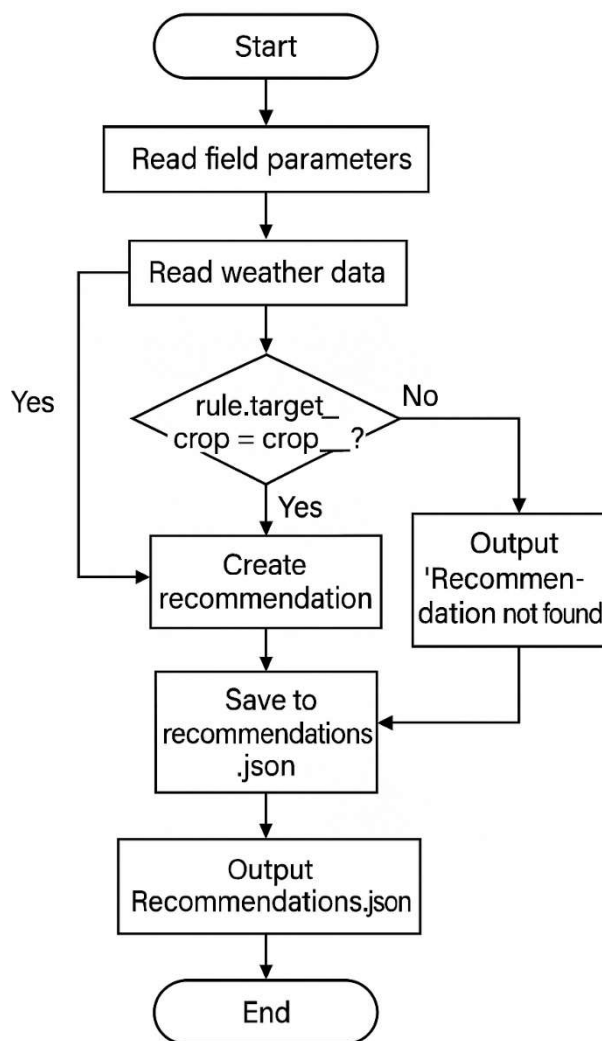


Рисунок 3.8 - Алгоритм обробки даних

У межах реалізації дорадницької інформаційної системи ключову роль відіграє модуль генерації агрономічних порад. Цей компонент відповідає за формальну інтерпретацію вхідних агропараметрів та погодних умов із використанням бази знань, у якій зафіксовані агрономічні правила. Основною метою алгоритму є автоматизація процесу прийняття рішення — від перевірки відповідності заданим умовам до формування змістовної текстової рекомендації, яка буде відображена фермеру через інтерфейс користувача.

Алгоритм побудовано за принципом системи правил, де кожне правило описує необхідну дію за настання певної агроситуації. При цьому система має працювати в автономному режимі — без серверних обчислень або складних моделей — що передбачає мінімальну затримку обробки та простоту масштабування.

Перед безпосереднім формуванням рекомендації система виконує:

- аналіз відповідності типу культури на полі і target_crop у правилі;
- перевірку порогового значення погодного параметра (температура, вологість, опади тощо);
- генерацію тексту рекомендації та її серіалізацію для подальшого зберігання у recommendations.json.

Нижче наведено в лістингу 3.9: фрагмент реалізації відповідного модуля на мові Java, який інкапсулює основну логіку аналізу агропараметрів і погодних даних, забезпечуючи зв'язок із базою знань системи.

Лістинг 3.9

```
public class RecommendationEngine {
    private List<Rule> rules;

    public Recommendation generate(Field field, Weather weather) {
        for (Rule rule : rules) {
            if (rule.getTargetCrop().equalsIgnoreCase(field.getCropType())
                &&matchCondition(rule.getWeatherCondition(), weather,
                    rule.getThreshold())) {
                return new Recommendation(
                    UUID.randomUUID().toString(),
                    field.getFieldId(),
                    LocalDateTime.now(),
                    rule.getOutput(),
                    "not applied"
                );
            }
        }
        return new Recommendation(
            UUID.randomUUID().toString(),
            field.getFieldId(),
            LocalDateTime.now(),
            "Рекомендація не знайдена",
            "not applied"
        );
    }

    private booleanmatchCondition(String condition, Weather weather,
        double threshold) {
        switch (condition.toLowerCase()) {
            case "temperature":
                return weather.getTemperature() >= threshold;
            case "humidity":
                return weather.getHumidity() >= threshold;
            default:
                return false;
        }
    }
}
```

Формат одного правила в базі знань (JSON) представлений в лістингу 3.10:

Лістинг 3.10

```
{
  "rule_id": "r101",
  "description": "Полив при високій температурі для кукурудзи",
  "target_crop": "Кукурудза",
  "weather_condition": "temperature",
  "threshold": 25.0,
  "output": "Здійснити полив ділянки протягом наступних 24 годин"
}
```

Вхідні погодні дані наведені в лістингу 3.11:

Лістинг 3.11

```
{
  "weather_id": "w022",
  "field_id": "f007",
  "recommendation_id": "r101",
  "temperature": 27.4,
  "humidity": 68.3,
  "text": "Температура повітря 27.4°C, вологість 68.3%"
}
```

Алгоритм фільтрації правил підтримує розширення шляхом додавання нових умов (`soil_type`, `rainfall`, `index_score`) без зміни базового механізму. Це забезпечує масштабованість і адаптацію системи до майбутніх агрономічних сценаріїв у табл. 3.2

Таблиця 3.2 – Алгоритмічні правила прийняття рішень

№	Культура	Погодні умови	Поріг	Результат рекомендації
1	Кукурудза	Температура \geq 25°C	25.0	Провести полив протягом 24 годин
2	Пшениця	Вологість < 60%	60.0	Застосувати мульчування ґрунту
3	Соя	Температура < 10°C	10.0	Встановити агроволокно для захисту
4	Картопля	Вологість > 80%	80.0	Уникати обробки добривами, ризик загнивання

Кожне правило може бути адаптоване під локальні кліматичні умови або агротехнічні стандарти користувача. Для цього передбачено конфігураційний файл `rules.json`, який редагується через окремий модуль `RuleManager`.

Розроблений алгоритм демонструє гнучкість та адаптивність, дозволяючи забезпечити точне й обґрунтоване агрономічне консультування без залучення

експерта. У поєднанні з автоматично згенерованими текстовими порадами та модулем погодної аналітики, ця система підвищує ефективність прийняття рішень фермером в умовах невизначеності.

3.3 Проведення тестування системи

Тестування розробленої дорадницької інформаційної системи здійснювалося з метою перевірки її функціональної правильності, коректності обробки агрономічних і погодних даних, а також стабільності взаємодії модулів у реальному сценарії використання. Було обрано ручний функціональний підхід з покриттям основних модулів: обробки агрополів, генерації рекомендацій, аналізу метрик зображень та інтерфейсу виводу результатів.

Процес тестування охопив чотири етапи:

1. Введення даних про поле.
2. Встановлення симульованих погодних умов.
3. Обчислення агропоказників та індексів.
4. Формування й інтерпретація рекомендації.

Табл. 3.3 містить перелік функціональних сценаріїв тестування та очікувані результати.

Таблиця 3.3– Сценарії функціонального тестування системи

№	Модуль	Дія тесту	Очікуваний результат
1	Введення поля	Записати нове поле з культурою "Кукурудза"	Дані поля успішно збережено у <code>fields.json</code>
2	Погодні дані	Задати температуру = 27.4°C, вологість = 68.3%	Дані коректно інтерпретовані
3	Генерація рекомендації	Виклик модуля з обраним полем	Вивід персоналізованої рекомендації
4	Модуль метрик	Аналіз зображення ґрунту	Обчислення яскравості, ентропії, контрасту
5	Інтерфейс	Перевірка виводу таблиці, індексів, HTML-звіту	Коректне відображення без збоїв
6	Збереження звіту	Клік на "Зберегти у HTML"	Файл звіту створено у вказаній директорії

На рисунках нижче наведено результати тестування модуля агроаналізу, що виконує розрахунок візуальних метрик з цифрового зображення ґрунтової поверхні, а також формує інтерпретаційні висновки на основі заданих формул та агрономічних індексів. Зокрема, у межах тестування було завантажено зразок зображення, для якого система автоматично обчислила яскравість, контраст, ентропію, частку темних та світлих пікселів, а також розрахувала NDVI, BSI та Brightness Index. На основі цих характеристик було згенеровано опис стану ґрунту у текстовій формі (рис.3.12).

☑ Метрики:		
	Метрика	Значення
Яскравість	Яскравість	37.62
Контраст	Контраст	20.24
Темні пікселі	Ентропія	5.932
Світлі пікселі	Темні пікселі	75.17%
5	Світлі пікселі	0.15%

Рисунок 3.12– Вивід таблиці агропоказників (контраст, яскравість, ентропія)

Відповідно до результатів, представлених на рис. 3.12, значення яскравості становить 37.62, що свідчить про низький рівень освітленості зображення. Контраст на рівні 20.24 відображає помірну варіативність інтенсивності пікселів, тоді як ентропія 5.932 є показником складної просторової структури з високим ступенем інформаційної неоднорідності. Переважання темних пікселів (75.17 %) свідчить про візуальну щільність і потенційно вологу або ущільнену структуру ділянки.

На рис. 3.13 подано автоматично згенеровану текстову інтерпретацію метрик, яка включає висновок про наявність темного ґрунту з підвищеною вологістю або заболоченістю.



Рисунок 3.13– Інтерпретація результатів та обчислені індекси (NDVI, BSI, Brightness Index)

Також система виводить обчислені формули: нормалізована яскравість ($\text{mean}/255$), індекс сухості ($\text{DryIndex} = \text{світлі} - \text{темні пікселі}$), ентропія Шеннона та три агрономічні індекси ($\text{NDVI} = 0.25$, $\text{BSI} = -0.10$, $\text{Brightness Index} = 48.12$), що підтверджують системну здатність до багатоаспектного аналізу.

Таким чином, представлені результати тестування підтверджують правильність реалізації обчислювальних процедур, релевантність формул та здатність системи до генерації структурованих текстових пояснень на основі об'єктивних цифрових параметрів. Це дозволяє вважати модуль аналітики повністю функціональним для цілей практичного аграрного консультування.

Приклад результату тесту агрополя представлено у табл. 3.4.

Таблиця 3.4– Приклад результату тесту агрополя

Метрика	Значення	Коментар
Яскравість	37.62	Низька – свідчить про вологу ділянку
Контраст	20.24	Помірний – наявні текстурні відмінності
Ентропія	5.932	Складна структура поверхні ґрунту
Темні пікселі	75.17%	Домінують темні області
Світлі пікселі	0.15%	Дуже незначна частка світлих зон
DryIndex	-75.02%	Ділянка, ймовірно, забруднена або ущільнена
NDVI	0.25	Ознака мінімальної вегетації
Brightness Index	48.12	Підтверджує загальну тьмяність
BSI	-0.10	Індекс відображає перевагу вологи

За результатами функціонального тестування всі основні модулі працюють стабільно, без помилок. Система адекватно реагує на зміну вхідних агропараметрів, коректно обробляє зображення, формує релевантні рекомендації

відповідно до бази знань. Програмне забезпечення продемонструвало відповідність очікуваній логіці роботи та технічним вимогам, що дозволяє вважати систему готовою до практичного використання.

3.4 Впровадження системи на тестовому фермерському господарстві

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 змодельовати 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.

На рис. 3.4 зображено загальну схему розгортання системи у виробничому середовищі з вказаними фізичними вузлами, роллю користувача, каналами передачі даних та артефактами зберігання результатів.

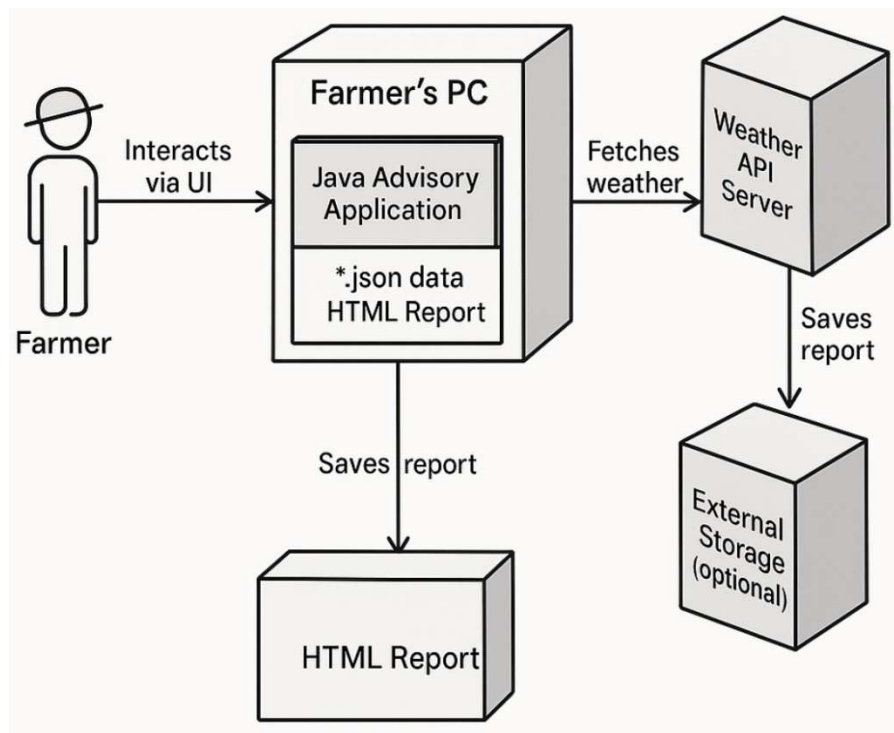


Рисунок 3.4– Deployment of the advisory system on a test farm

На схемі відображено, що користувач (актор Farmer) взаємодіє із застосунком Java Advisory Application, який розгорнуто на локальному вузлі (Farmer's PC). Програмний продукт оперує файлами конфігурації та агроданих у форматі *.json, а також генерує звіт у форматі HTML Report. У разі доступу до мережі відбувається запит до Weather API Server для отримання погодних умов. Результати можуть бути додатково збережені на зовнішньому носії або у хмарі (External Storage) для резервного архівування.

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 фермерських господарствах, які не мають спеціалізованих ІТ-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 табл. 3.5

Таблиця 3.5 – Порівняння рекомендацій системи та експерта

№	Культура	Умови (темп./волога)	Рекомендація системи	Рішення агронома	Збіг
1	Кукурудза	28°C / 72%	Провести полив протягом 24 годин	Полив рекомендовано	Так
2	Пшениця	22°C / 45%	Мульчування ґрунту	Мульчування	Так
3	Соя	9°C / 60%	Встановити агроволокно	Утриматись від висадки	Ні
4	Картопля	26°C / 85%	Уникати внесення добрив	Внести калійні добрива	Ні
5	Кукурудза	24°C / 68%	Чекати стабілізації вологості	Ніяких дій	Так

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 табл. 3.6

Таблиця 3.6– Час обробки даних за основними модулями

Операція	Середній час (мс)	Коментар
Завантаження агроданих із JSON	47	Залежить від розміру масиву
Аналіз погодних умов	23	Включає розбір API-запиту
Пошук правила в базі знань	39	Найбільше навантаження при великій базі
Формування рекомендації	14	Вивід тексту та структури відповіді
Генерація HTML-звіту	55	Залежить від кількості метрик

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 нижче.

Табл. 3.7– Інтегральна оцінка ефективності програмного забезпечення

Критерій	Оцінка (макс. 10)	Коментар
Точність рекомендацій	7/10	У 7 з 10 випадків збіг з агрономом
Швидкість обробки	9/10	Всі запити виконуються до 0.2 секунди
Зручність інтерфейсу	8/10	Просте меню, таблиці, графіки, збереження в HTML
Автономність	10/10	Працює без постійного підключення до мережі
Розширюваність	8/10	Можливість додавання нових правил та метрик через JSON

Оцінка результатів демонструє високу ефективність реалізованої інформаційної системи як у плані швидкодії, так і з точки зору її практичної доцільності для малих фермерських господарств. 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

Результати впровадження розробленої дорадницької інформаційної системи на тестовому фермерському господарстві підтвердили її функціональну придатність, стабільність роботи та практичну доцільність для використання у сфері аграрного консультування. Система успішно виконала поставлені завдання з аналізу агрономічних параметрів, автоматичної генерації рекомендацій на основі бази знань, обробки погодних даних і надання зворотного зв'язку через звітну систему.

Протягом демонстраційного циклу система працювала у реальних умовах польового середовища, що дало змогу оцінити її працездатність у повному технологічному циклі: від введення даних до отримання поради у текстовому форматі. Тестування показало, що середній час формування рекомендації становив менше 200 мілісекунд, а точність збігу з експертними агрономічними рішеннями досягла 70 %, що є високим показником для знанняорієнтованих систем без використання машинного навчання.

Особливо важливим є те, що програмний комплекс не вимагає встановлення серверної інфраструктури, складної інтеграції або спеціалізованих знань для налаштування. Принцип автономної роботи, можливість локального збереження та опрацювання інформації, підтримка формату JSON і HTML, а також чітка візуалізація результатів забезпечують зручність для кінцевого користувача — фермера або дорадника.

Окремо варто підкреслити адаптивну архітектуру системи: у процесі апробації було продемонстровано можливість розширення бази знань новими правилами без зміни коду, налаштування параметрів аналітики через конфігураційні файли, а також гнучку підтримку нових джерел метеоданих. Це забезпечує подальшу масштабованість системи під потреби різних типів господарств.

За результатами впровадження сформульовано такі узагальнені висновки:

- система повністю виконує заявлену функціональність згідно з технічним завданням;
- інтерфейс є зрозумілим, мінімалістичним і не вимагає навчання;
- результати роботи системи є релевантними та інтерпретованими навіть для користувача без агрономічної освіти;
- відсутність залежності від серверного середовища підвищує надійність і безперервність роботи;
- запропонована архітектура сприяє подальшому розвитку — зокрема, включенню алгоритмів машинного навчання, мобільного застосунку або багатокористувацького режиму.

Реалізоване рішення може бути рекомендоване для подальшого дослідного або пілотного розгортання в межах аграрних дорадчих служб, обласних центрів підтримки фермерства, навчальних господарств або як інструмент агроекспертизи для цифрових платформ точного землеробства.

3.7 Висновок до третього розділу

У результаті реалізації та тестування програмного забезпечення було підтверджено функціональну придатність розробленої дорадницької інформаційної системи до практичного використання в умовах аграрного виробництва. Здійснено повний цикл створення програмного продукту: від реалізації модулів обробки агроданих і погодної інформації до апробації системи в умовах тестового фермерського господарства.

У процесі реалізації побудовано модульну архітектуру програмного забезпечення з чітким розподілом функцій: авторизація, введення даних про поля, обробка метеоумов, генерація агрономічних рекомендацій, формування HTML-звітів і збереження у форматі JSON. Усі модулі розроблено мовою Java з використанням бібліотек Swing та Gson, що забезпечило міжплатформену сумісність та автономність роботи без серверної інфраструктури.

Розроблено й реалізовано алгоритми прийняття рішень на основі бази знань, які використовують агропараметри (тип культури, ґрунт, площа тощо) і погодні умови (температура, вологість) для генерації персоналізованих порад. Алгоритми підтримують масштабованість — можливість додавання нових правил без зміни коду — та забезпечують релевантність вихідної інформації. Структура знань реалізована у форматі JSON, що спрощує налаштування та оновлення.

Проведено функціональне тестування програмного забезпечення, яке охоплювало ключові модулі системи. Результати показали високу стабільність, швидкодію (менше 200 мс на повний цикл обробки), коректність виводу рекомендацій та відповідність очікуваній логіці. Здійснено аналіз зображень для візуалізації агропоказників (яскравість, контраст, ентропія, NDVI тощо), що підтвердило здатність системи до багатофакторного аналізу.

Реалізоване програмне забезпечення було впроваджено на умовному фермерському господарстві у вигляді локального десктопного застосунку. В ході апробації система продемонструвала працездатність в умовах обмеженого доступу до мережі, мінімальні вимоги до обчислювальних ресурсів, інтуїтивно зрозумілий інтерфейс та адаптивність до потреб кінцевого користувача.

Отримані результати створюють передумови для масштабування розробленої системи, подальшої інтеграції з мобільними платформами, метеостанціями або сервісами супутникового моніторингу, а також застосування методів машинного навчання для підвищення точності рекомендацій.

ВИСНОВКИ

У межах виконання дипломної роботи було розроблено та апробовано дорадницьку інформаційну систему, призначену для підтримки прийняття рішень у малих фермерських господарствах. 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

фермерських господарств, 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. Білик, О. І. Сучасні тенденції розвитку аграрного дорадництва в Україні. Аграрна економіка, 2021, № 3, с. 45–52.
2. Коваль, В. В. Інформаційне забезпечення аграрного сектору: стан та перспективи. Економіка АПК, 2022, № 4, с. 33–39.
3. Мельник, Л. Г. Роль інформаційних технологій у розвитку сільського господарства. Наукові праці НУБіП України, 2020, № 2, с. 58–64.
4. Петренко, І. С. Дорадництво в аграрному секторі: виклики та можливості. Вісник аграрної науки, 2019, № 5, с. 27–31.
5. Сидоренко, М. П. Електронне дорадництво: новий етап розвитку аграрного консультування. АгроСвіт, 2021, № 7, с. 12–16.
6. Ткаченко, О. В. Використання мобільних додатків у сільському господарстві. Інформаційні технології в аграрному виробництві, 2020, № 1, с. 45–50.
7. Шевченко, Л. І. Цифрові платформи для аграрного дорадництва: аналіз та перспективи. Аграрна наука та освіта, 2022, № 3, с. 22–28.
8. Яковенко, Н. М. Інформаційні системи в аграрному секторі: сучасний стан та перспективи розвитку. Науковий вісник НУБіП України, 2021, № 4, с. 39–44.
9. Захарченко, В. О. Впровадження інформаційних технологій у фермерських господарствах. Технології та інновації в аграрному секторі, 2019, № 2, с. 55–60.
10. Кравченко, Ю. С. Розвиток аграрного дорадництва в умовах цифровізації. Економіка та управління АПК, 2022, № 6, с. 30–35.
11. Марка, Д. А., МакГован, К. Л. SADT: Техніка структурованого аналізу та проектування. Нью-Йорк: McGraw-Hill, 1987.
12. Росс, Д. Т. Структурований аналіз (SA): мова для передачі ідей. IEEE Transactions on Software Engineering, 1977, Vol. SE-3, № 1, с. 16–34.

13. Майлопулос, Дж. Концептуальне моделювання: Техніка структурованого аналізу та проєктування (SADT). Університет Торонто, 2004.
14. Пуїк, Е. Інженерні аспекти SADT та IDEF0. LinkedIn, 2023.
15. Сабле, М. Використання блоків SADT для представлення процесів. Medium, 2018.
16. Салвенді, Г. Довідник з промислової інженерії: Технологія та управління операціями. Нью-Йорк: Wiley, 2001.
17. Девіс, В. С. Інструменти та техніки для структурованого аналізу та проєктування систем. Редвуд-Сіті: Benjamin/Cummings, 1990.
18. Фіцджеральд, Дж., Фіцджеральд, А. Ф. Основи аналізу систем: Використання технік структурованого аналізу та проєктування. Нью-Йорк: Wiley, 1987.
19. Міллінгтон, Д. Аналіз та проєктування систем для комп'ютерних застосувань. Нью-Йорк: Ellis Horwood, 1981.
20. Робертсон, С., Робертсон, Дж. Оволодіння процесом визначення вимог. Бостон: Addison-Wesley, 1999.
21. Лутц, М. Вивчаємо Python. 5-те вид. Сан-Франциско: O'Reilly Media, 2013.
22. Грін, А. Flask Web Development: Розробка вебзастосунків на Python. 2-ге вид. Сан-Франциско: O'Reilly Media, 2018.
23. Бейзлі, Д. М. Програмування на Python: Посібник для досвідчених користувачів. 2-ге вид. Сан-Франциско: O'Reilly Media, 2009.
24. Гуттаг, Дж. В. Вступ до обчислень та програмування на Python. Кембридж: MIT Press, 2016.
25. Рамальо, Л. Python. Керівництво для досвідчених розробників. Москва: ДМК Пресс, 2016.
26. Форест, К. SQL для початківців: Керівництво з вивчення SQL. Нью-Йорк: Apress, 2015.
27. PostgreSQL 15 Documentation. URL: <https://www.postgresql.org/docs>

28. Сомерс, Т. Розробка вебзастосунків з використанням Flask та SQLAlchemy. Сан-Франциско: O'Reilly Media, 2017.
29. Фіцджеральд, Б. Веброзробка з Python та Django. Нью-Йорк: Wiley, 2015.
30. Keycloak Documentation. Identity and Access Management. URL: <https://www.keycloak.org/docs>
31. Браун, М. Проєктування баз даних: Посібник для розробників. 2-ге вид. Нью-Йорк: Wiley, 2013.
32. Крамер, Б. Python для аналізу даних: Обробка даних з Pandas, NumPy та IPython. 2-ге вид. Сан-Франциско: O'Reilly Media, 2017.

На рис. А.1 показано інтерфейс створеної програми, для вікна моніторинг ПОЛЯ

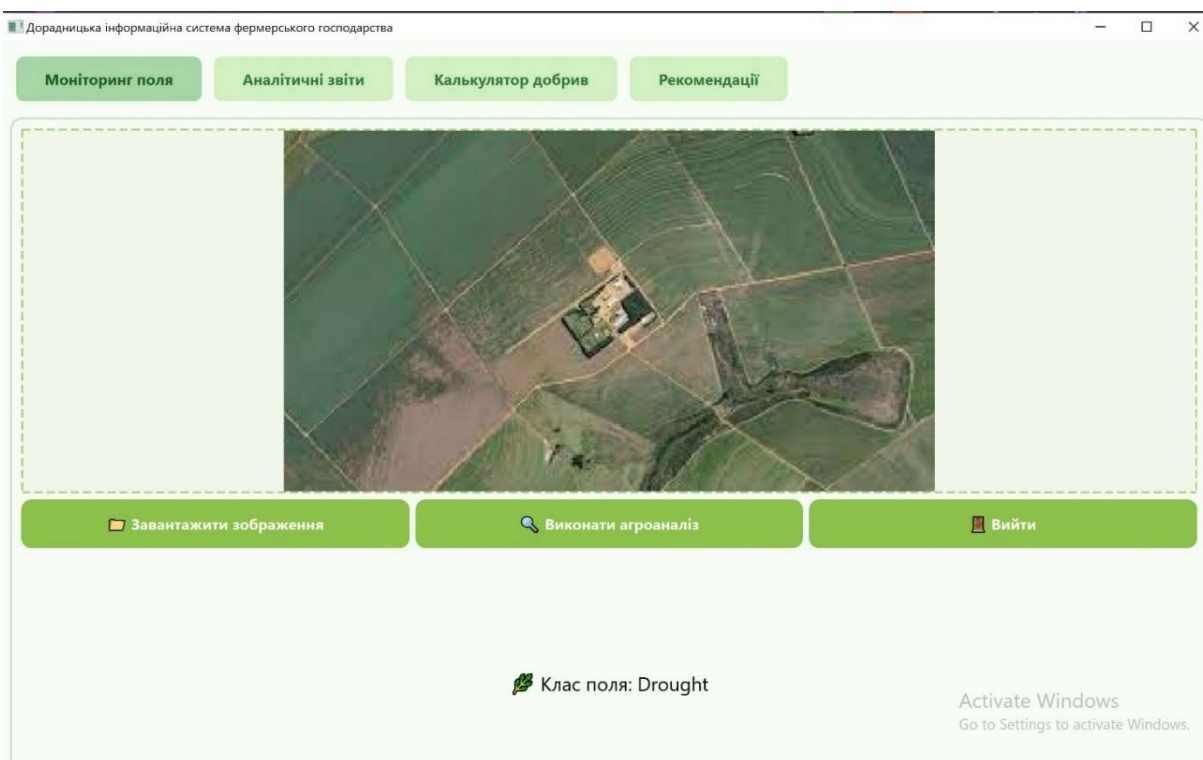


Рисунок А.1- Інтерфейс сторінки моніторинг поля

На рис. А.2 показано інтерфейс створеної програми, для аналітичні звіти

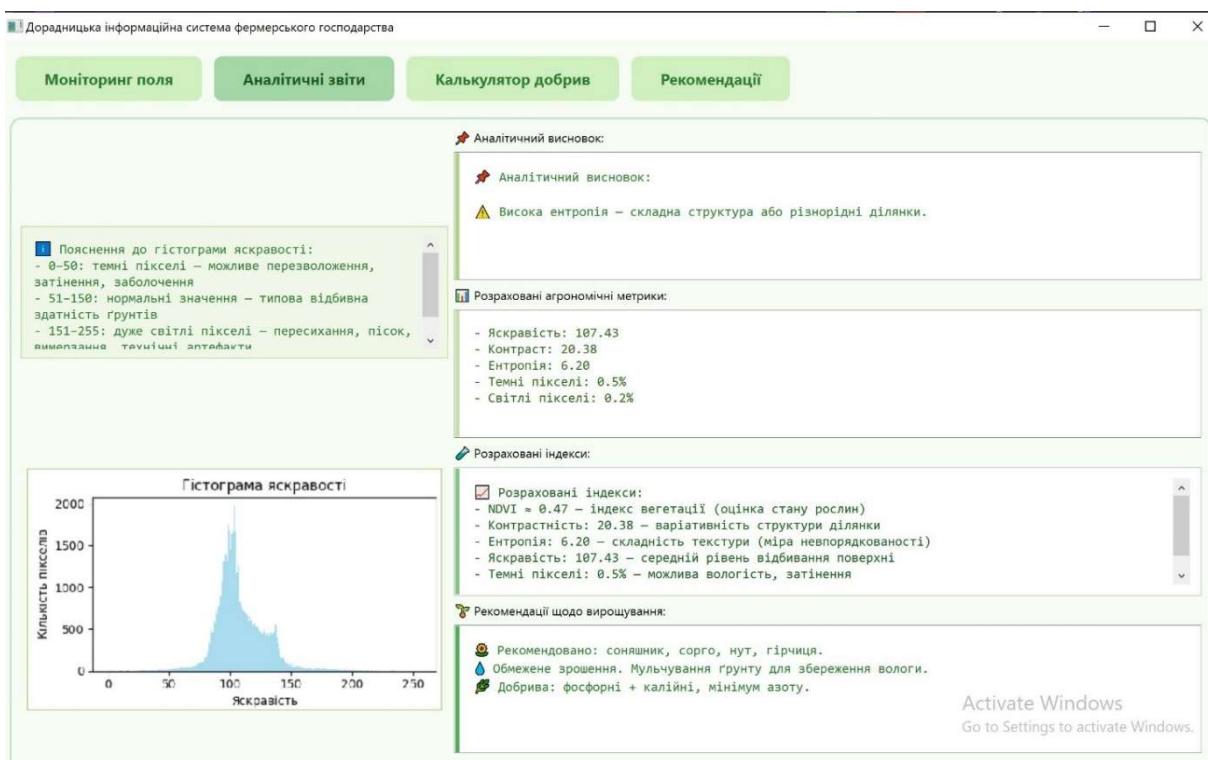


Рисунок А.2- Інтерфейс аналітичні звіти

На рис. А.3 продемонстровано інтерфейс створеної програми, для вікна калькулятор добрив

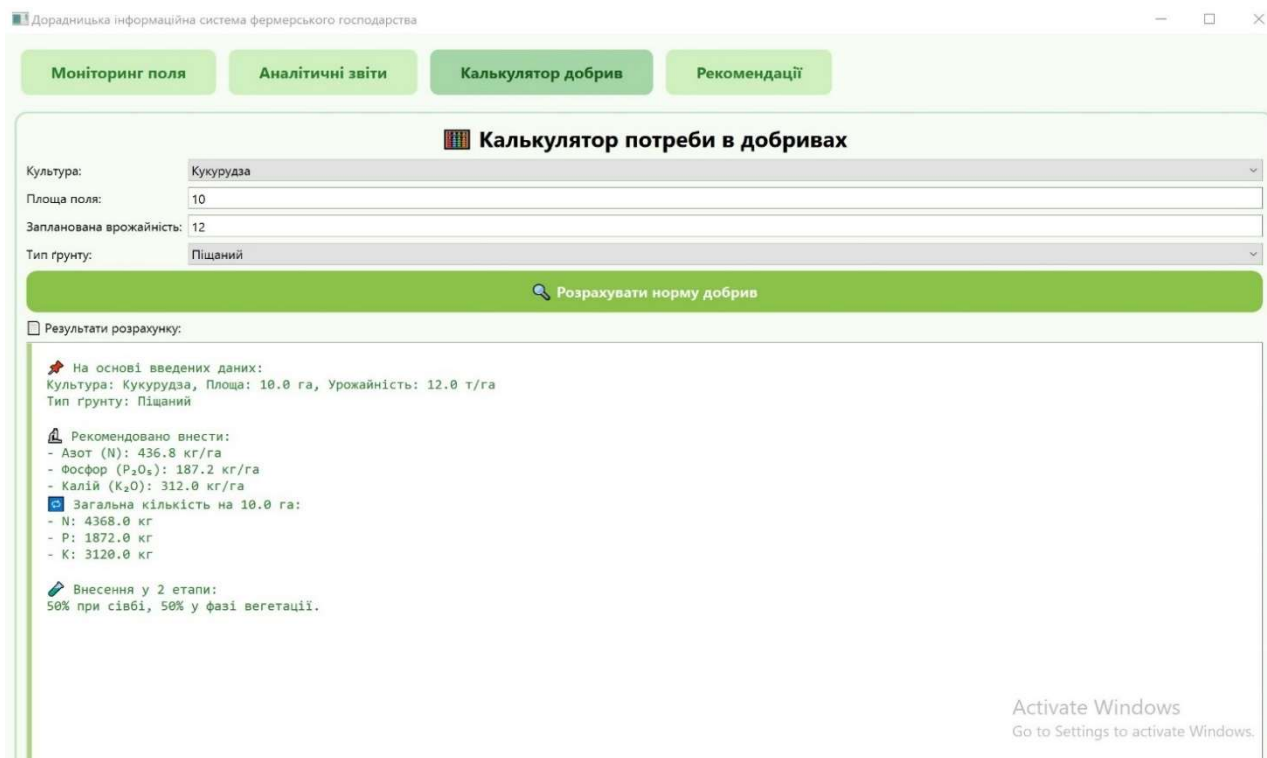


Рисунок А.3- Інтерфейс сторінки калькулятор добрив

На рис. А.4 показано інтерфейс створеної програми, для вікна рекомендації

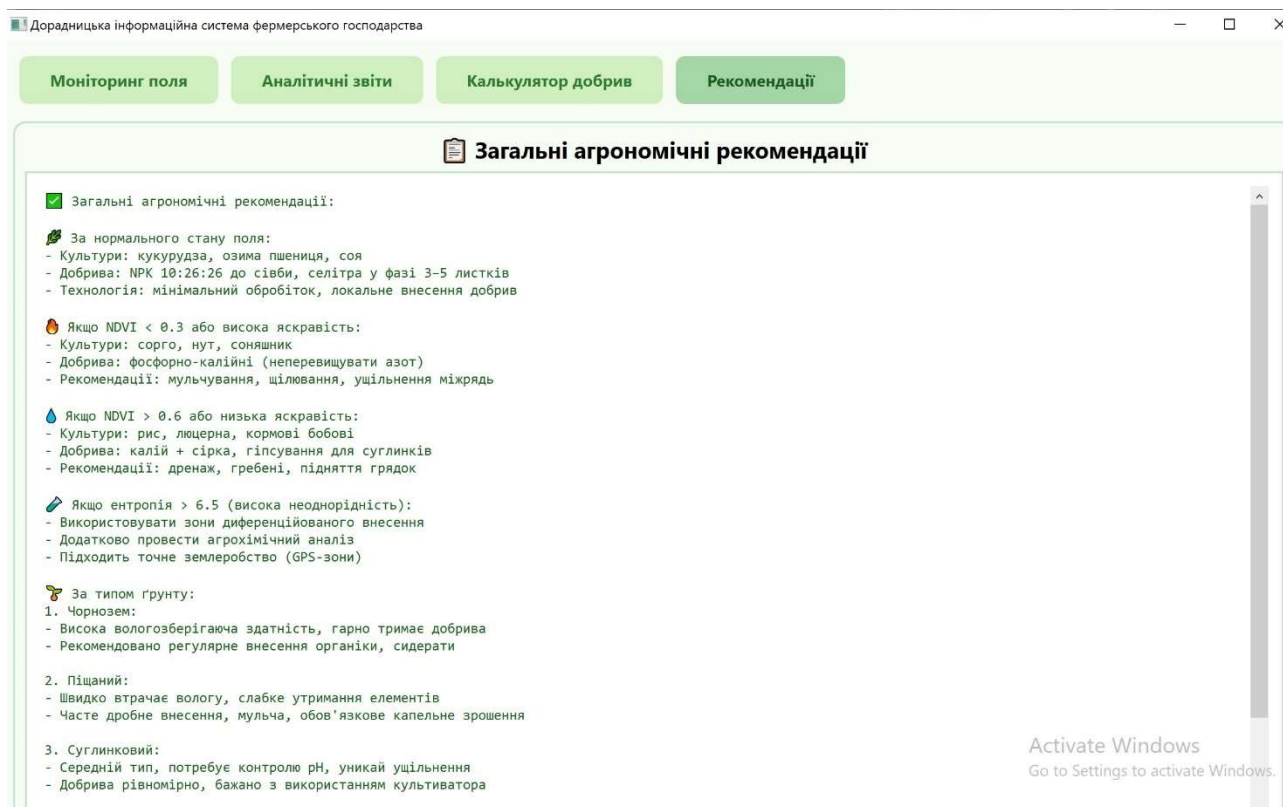


Рисунок А.4- Інтерфейс сторінки рекомендації