

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
Факультет інформаційних технологій

УДК 004.9:631.559

«ПОГОДЖЕНО» «ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»  
НУБІП України НУБІП України  
Декан Завідувач  
факультету інформаційних кафедри комп'ютерних наук  
технологій

д.пед.н., професор Газаунова О.Г. к.т.н., доцент Голуб Б.Л.  
« » 20 р « » 20 р

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА  
на тему «Система підтримки прийняття рішень забезпечення врожайності с/г культур агрофірми»  
Спеціальність 121 Інженерія програмного забезпечення  
(шифр і назва)

Освітньо-професійна програма Програмне забезпечення інформаційних систем  
виробнича, дослідницька  
Орієнтація освітньої програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми / Голуб Б.Л. /  
к.т.н., доцент (науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи / Бородкін Г.О. /  
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Науковий консультант магістерської кваліфікаційної роботи / Бородкіна І.Л. /  
к.т.н., доцент (науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав / Зах В.Ю. /  
(підпис) (ПІБ студента)

НУБІП КИЇВ-2021 України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**  
Факультет (ННІ)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри \_\_\_\_\_

\_\_\_\_\_  
(науковий ступінь, вчене звання) (підпис) (П.І.Б.)  
" " 20 року

**ЗАВДАННЯ**

**ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ  
ЗАХУ ВАДИМУ ЮРІЙОВИЧУ**

Спеціальність 121 – «Інженерія програмного забезпечення»

Освітня програма «Програмне забезпечення інформаційних систем»

Орієнтація освітньої програми - освітньо-професійна

Тема магістерської кваліфікаційної роботи – «Система підтримки прийняття рішень забезпечення врожайності с/г культур агрофірми»

затверджена наказом ректора НУБіП України від "29" жовтня 2020р. №1636"С"

Термін подання завершеної роботи на кафедру 2021 рік 11 місяць 30 число

Вихідні дані до магістерської кваліфікаційної роботи: застосування сучасних методів та технологій проектування для створення системи підтримки прийняття рішень забезпечення врожайності с/г культур агрофірми

Перелік питань, що підлягають дослідженню.

Системний аналіз предметної області;

Проектування програмного забезпечення;

Опис програмної реалізації;

Розгортання програмного додатку;

Перелік графічного матеріалу (за потреби)

Дата видачі завдання "12" листопада 2020 р.

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_

Бородкін Г. О.

Завдання прийняв до виконання \_\_\_\_\_

Зах В. Ю.

# НУБІП України **1. ЗМІСТ**

Зміст

3

# НУБІП України

Перелік умовних позначень

5

Вступ

6

Розділ 1 9

1.1 9

# НУБІП України

1.2 11

1.3 12

1.4 16

# НУБІП України

Розділ 2 19

2.1 19

2.2 21

2.3 24

# НУБІП України

2.4 26

Розділ 3 28

3.1 28

# НУБІП України

3.2 31

3.3 32

3.4 35

# НУБІП України

3.5 36

3.6 40

Розділ 4 44

4.1 44  
4.2 47  
4.3 50  
НУБІП України

4.4 53  
ВИСНОВКИ 57  
Список використаних джерел 59  
НУБІП України

ДОДАТОК А

61

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

# НУБІП України

## 2. ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AWS – веб сервіси компанії Amazon, Amazon Web Services

CDN – мережа розповсюдження контенту, Content Delivery Network

JDK – комплект розробника застосунків, Java Development Kit

JRE – середовище виконання Java, Java Runtime Environment

JVM – віртуальна машина Java, Java Virtual Machine

MVC – архітектурний шаблон “Модель-вигляд-контролер”

REST – передача репрезентативного стану, Representative State Transfer

SPA – односторінковий інтерфейс, Single Page Application

SVG – масштабована векторна графіка, Scalable Vector Graphics

UML – уніфікована мова моделювання, Unified Modeling Language

ПЗ – програмне забезпечення

# НУБІП України

# НУБІП України

# НУБІП України

# НУБІП України

# НУБІП України

## 3. ВСТУП

**Актуальність.** Сільське господарство є галуззю господарства, яке спрямоване на забезпечення населення провізією та отримання сировини для різних галузей виробництва. Для оптимізації процесів виробництва, зменшення ризиків та покращення процесів планування сівозмін можна використовувати системи підтримки прийняття рішень.

Системи підтримки прийняття рішень є такими засобами, які за допомогою збирання, обробки та аналізу інформації здатні впливати на процеси ухвалення управлінських рішень в господарстві. Використання систем такого типу дозволяє підвищити врожайність сільськогосподарських культур агрофірми за рахунок більш ефективного планування та управління господарством. Це призводить до підвищення врожайності та зменшення собівартості виробленого продукту, що надає агрофірмі, яка використовує системи підтримки прийняття рішень, конкуренту перевагу над іншими аграріями, які використовують більш традиційний підхід до процесу прийняття рішень.

Основним завданням розробленої у рамках цієї роботи системи є підвищення ефективності короткострокового та довгострокового планування сівозмін сільськогосподарських культур.

**Актуальність** розробки системи підтримки прийняття рішень забезпечення врожайності в тому, що використання засобів такого типу є інтенсивним методом розвитку господарства, який дозволяє підвищити ефективність, зменшити адміністративні витрати на управління агрофірмою та скоротити виснаження земель. Останній фактор є особливо актуальним на даний момент в Україні, через активне розширення площ зайнятих соняшником, який потребує великої кількості

# НУБІП УКРАЇНИ

поживних речовин і виснажує землі, що потребує додаткової уваги до планування сівозмін.

В Україні керівництво агрофірм не надає значної уваги системам такого типу.

Причинами цієї ситуації є обмежені можливості існуючих вітчизняних систем, та висока ціна закордонних реалізацій. Можна зробити висновок, що для забезпечення інтенсивного розвитку сільськогосподарської галузі в Україні необхідно залучати інформаційні технології.

# НУБІП УКРАЇНИ

**Об'єкт дослідження** – процес прийняття управлінських рішень для забезпечення врожайності.

**Предмет дослідження** – система підтримки прийняття рішень забезпечення врожайності с/г культур агрофірми.

# НУБІП УКРАЇНИ

**Мета** – на основі використання серверних технологій створити комп'ютерну систему підтримки процесу планування сівозмін для фермерських господарств.

## Завдання:

1. Дослідити предметну область, яка пов'язана з плануванням сівозмін.
2. Проаналізувати існуючі рішення з плануванням сівозмін та запропонувати варіант реалізації у системі.
3. Запропонувати свій варіант реалізації системи підтримки прийняття рішень забезпечення врожайності сільськогосподарських культур агрофірми на основі впровадження технології плануванням сівозмін.
4. Додати аналітичний та статистичний модулі до розробленого раніше програмного додатку обліку врожайності пелів.

# НУБІП УКРАЇНИ

**Методи дослідження** – для проведення аналізу даних та зведення статистики використовується мова Java та фреймворк Spring як основа додатку, для візуалізації даних у вигляді графіків – використано модуль ngx-charts разом з Angular,



оброблювані дані зберігаються у реляційній базі даних PostgreSQL, та вносяться у цю базу за допомогою раніше розробленого додатку обліку врожайності сільськогосподарських культур агрофірми.

**Наукова новизна** – запропоновано удосконалення систем обліку характеристик, врожайності сільськогосподарських культур та стану полів, за рахунок додання автоматичного аналізу та порівняння історичних даних, особливостей поля з оптимальними умовами для кожної окремої сільськогосподарської культури.

**Апробація результатів дослідження** – XII Міжнародна науково-практична конференція молодих вчених «інформаційні технології: економіка, техніка, освіта» - публікація «Система підтримки прийняття рішень забезпечення врожайності с/г культур агрофірми»

**Структура роботи** – робота у обсязі 63 сторінки, 25 рисунків, списку з 23 використаних джерел, одного додатку А, та складається з наступних частин:

Вступ

4 розділи.

1. Системний аналіз предметної області – дослідження методу підвищення врожайності за рахунок планування сівозмін.
2. Моделювання системи – дослідження та моделювання випадків використання системи користувачем.
3. Розробка системи – опис архітектури системи та реалізації окремих елементів функціоналу.
4. Результати дослідження – представлення результатів роботи та опис розробленого інтерфейсу користувача.

Висновки, Список використаних джерел

# РОЗДІЛ 1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис предметної області

Предметна область що автоматизується додатком – це процес прийняття рішень щодо планування сівозмін у агрофірмі.

На одній і тій же ділянці недоцільно декілька років підряд сіяти одну і ту ж культуру. Обов'язково повинна бути сівозміна. Це найкращий спосіб підвищення врожайності, боротьби з бур'янами шкідниками та хворобами [1].

Для аналізу та підбору найбільш пристосованої та вигідної з фінансової точки зору культури до теперішніх умов окремого поля, можна опиратись на наступні фактори:

Допустимість попередника для культури

- Чутливість до вирощування у повторних посівах
- Відповідність типу ґрунту до обраної сільськогосподарської культури
- Залежність культури від рівня кислотності ґрунту
- Вологість та зволоження ґрунту
- Строк повернення культури на попереднє поле
- Зональні характеристики поля
- Кількість необхідних речовин необхідних для окремої культури: азоту, калію, фосфору, кальцію, сірки, тощо.

Фінансова складова є також важливою при плануванні сівозмін. Наприклад, соняшник є однією з найбільш вигідних культур до вирощування в Україні, але він спричиняє виснаження земель і тому потребує особливої уваги до планування і оптимізації сівозмін.

Особливе важливим фактором при плануванні сівозмін є фактор попередника. При проектуванні схем динамічних сівозмін слід уникати чергування попередників, близьких між собою за біологією. Не рекомендується розміщувати колосові зернові після колосових, бобові після бобових [2].

Добрий попередник дозволяє стабільно отримувати підвищену врожайність наступної сільськогосподарської культури. Інші попередники можуть значно спотворювати умови для вирощування культури, призводячи до падіння врожаю та подальшого погіршення стану земель.

Але, деякі культури мають менш виражене відношення до вирощування у повторних посівах. За реакцією на повторне вирощування основні культури можна поділити на три групи: малочутливі, що практично не знижують урожайності при вирощуванні протягом кількох років на одному полі; середньочутливі та дуже чутливі, врожайність яких різко знижується навіть при вирощуванні протягом двох років на одному полі [3].

Опираючись на ці фактори можна з більшою точністю підібрати найбільш відповідну культуру, підвищити дохід агрофірми та ефективно протидіяти виснаженню земель.

Окремі фактори з цього переліку можна автоматично проаналізувати за допомогою програмного забезпечення, таким чином спростивши процес прийняття рішень для агрономів та керівників підприємств. Як джерело даних для аналізу можна використовувати розроблене попередньо рішення для обліку історичних даних полів. До можливості розширення функціоналу системи за допомогою додання параметрів обліку полів, можна додати можливість налаштування автоматичного аналізу цих параметрів системою при плануванні сівозмін.

НУБІП УКРАЇНИ

Ще одним з способів підвищення якості прийняття рішень агрофірмою є надання можливості агроному бачити статистику за параметрами для кожної окремої культури і поля. Це дозволить простежувати динаміку змін для полем та культурою, коректувати плани сівозмін й приймати рішення про додання добрив.

## НУБІП УКРАЇНИ

### 1.2 Аналіз вимог до програмної системи

До аналітичного і статистичного модулів програмної системи з обліку полів і підтримки прийняття рішень забезпечення врожайності с/г полів агрофірмою, висувуються наступні функціональні вимоги:

НУБІП УКРАЇНИ

1. Аналіз відповідності параметрів поля для окремої культури на основі історичних даних поля.

НУБІП УКРАЇНИ

2. Пошук найбільш оптимальної культури для обраного поля за існуючими даними на основі налаштованих рядків.
3. Підтримка аналізу цифрових параметрів на основі порівняння останніх історичних даних з поля з оптимальними значеннями для обраної

НУБІП УКРАЇНИ

- культури
4. Підтримка аналізу перелічуваних параметрів поля на основі таблиці відповідності параметру і його значення відносно обраної культури.
5. Отримання статистики у вигляді часових графіків на основі числових

НУБІП УКРАЇНИ

- параметрів для окремого поля та усіх культур, для окремого поля та окремої культури, та підтримкою часових проміжків.
6. Можливість конфігурувати рядків на основі числових параметрів через інтерфейс користувача.

НУБІП УКРАЇНИ

7. Можливість конфігурувати рядків на основі перелічуваних параметрів, та їх відношень відносно культури і значення перелічуваного параметра, за допомогою інтерфейсу користувача.

Також, до додатку висуваються наступні нефункціональні вимоги:

4. Відсутність історичних даних для будь-якого параметру не має призводити до помилки аналізу, так як параметри і радники можуть бути додані вже після вводу системи в експлуатацію.

5. Статистичний та аналітичний модулі мають бути додані до системи обліку полів та мати прямий доступ до бази даних.

6. Модулі мають бути побудовані на архітектурі MVC з використанням мови Java та технологій Spring, Angular.

7. Доступ до аналітики та статистики у режимі читання має бути доступним тільки для ролей агронома та керівника. Агрономом додатково має мати доступ до конфігурування радників у системі.

8. Усі функції системи мають бути доступні через веб-браузер.

### 1.3 Аналіз наявних рішень

Було розглянуто наступні програмні рішення в області аналізу та обліку стану сільськогосподарських полів:

#### OneSoil

OneSoil – безкоштовна платформа для землеробства, яка допомагає ефективно керувати полями. Платформа складається з мобільного додатка OneSoil Scouting та веб-додатку. Вони допомагають відстежувати зміни на полях, планувати сільськогосподарські роботи, підвищувати врожайність поля та економити ресурси

[4]

Переваги додатку **OneSoil**:

- Інтеграція з супутниковими системами
- Автоматична реєстрація полів з супутникових знімків на основі штучного інтелекту

# НУБІП України

- Аналіз врожайності на основі даних з комбайну
- Моніторинг стану рослин
- Використання є безкоштовним

Недоліки додатку **OneSoil**:

# НУБІП України

- Відсутні налаштування параметрів полів
- Обмежені можливості планування сівозмін
- Візуалізація змін у вигляді графіків не конфігурується

# НУБІП України

Приклад інтерфейсу додатку наведено на рис. 1.

**EOS Crop Monitoring**

Crop Monitoring дозволяє отримати максимально корисну інформацію на основі аналізу супутникових знімків. Це не тільки традиційний індекс вегетаційної рослинності NDVI, але й три інші вегетаційні індекси (NDRE, MSAVI, ReCI), оповіщення про зміну стану рослин, що налаштовуються, управління завданнями скаутингу, можливість зонування полів на основі показників вегетації, а також можливість створення карт-завдань для диференційованого внесення [5].

# НУБІП України

Переваги платформи **EOS**:

- Широка підтримка різних вегетаційних індексів
- Гарна підтримка супутникових інтеграцій
- Дуже широкі можливості візуалізації статистичних даних
- Новітній та функціональний інтерфейс

# НУБІП України

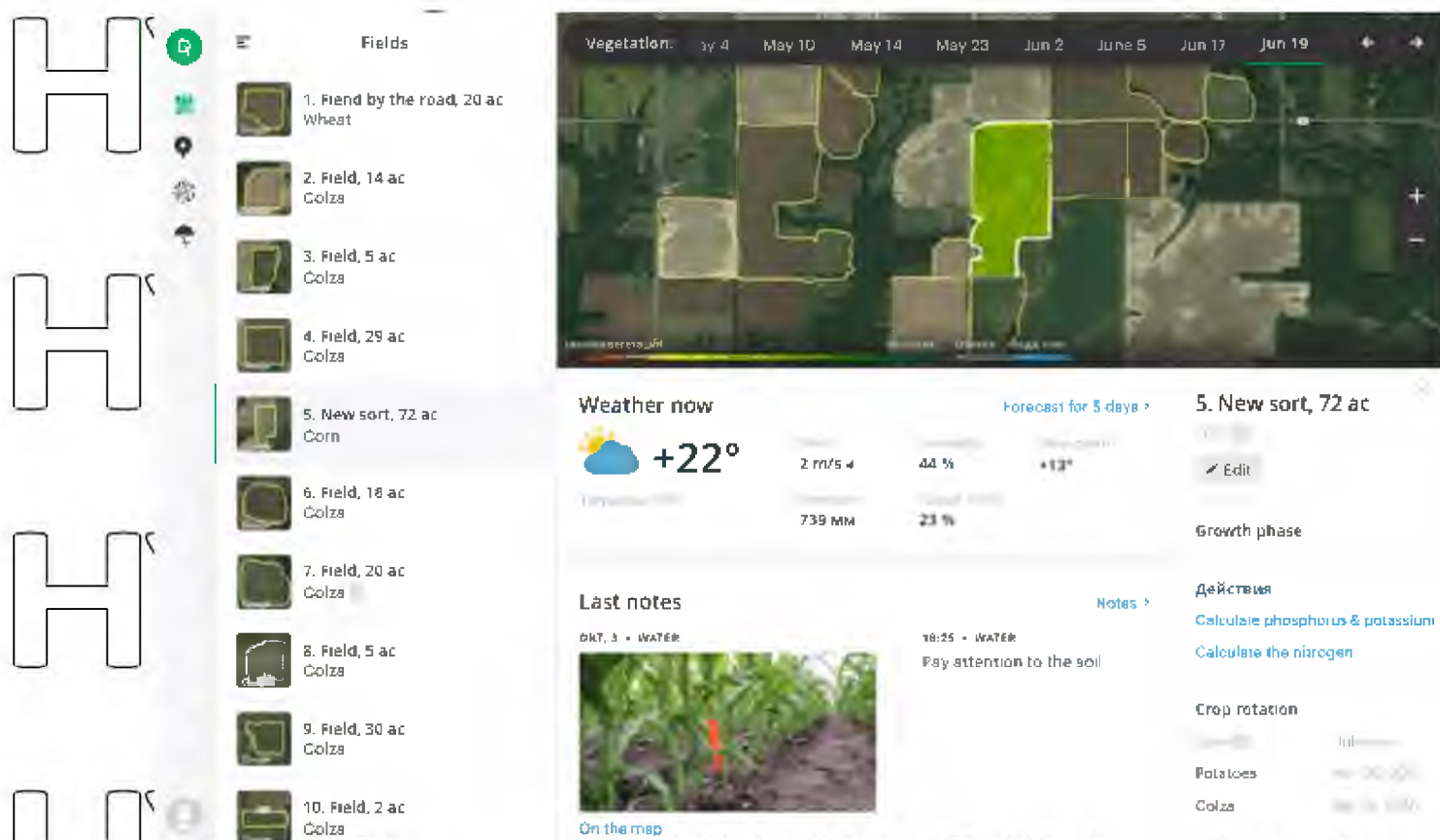


Рис. 1 Веб-інтерфейс додатку OneSoil

Медолги платформы EOS:

- Відсутні аналітичні можливості
- Складний процес внесення історичних даних
- Неможливо конфігурувати користувацькі параметри полів

Приклад інтерфейсу додатку EOS зображено на рис. 2



Рис. 2 Веб-інтерфейс додатку EOS

### Agremo

Agremo це стартап та продукт, що дозволяє клієнтам підраховувати рослини, шукати проблеми з шкідниками та бур'яном на основі інформації з дронів.

Сервіс спрощує моніторинг полів та дозволяє передбачати врожайність на основі історичних даних після.

### Переваги Agremo:

Активне залучення та інтеграція дронів до процесу моніторингу полів

- Пошук та передбачення захворювань рослин
- Аналітика врожайності
- Підтримка вегетаційних індексів полів

### Недолки Agremo:

- Обмежені статистичні можливості додатку



- Неможливо конфігурувати користувальницькі параметри полів
- Додаток все ще знаходиться у стані активної розробки

Приклад інтерфейсу додатку Agremo наведено на рис. 3.

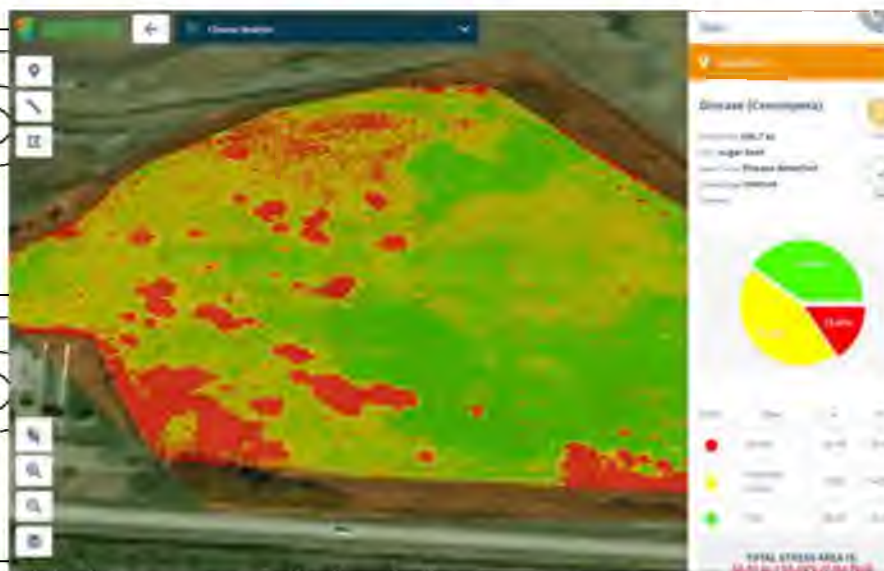


Рис. 3 Веб-інтерфейс додатку Agremo

#### 1.4 Постановка завдання

У рамках кваліфікаційної роботи було вирішено додати статистичний та аналітичний модулі до системи обліку врожайності полів.

Основна ідея системи є незмінною – система має надавати максимальну гнучкість агроному з боку налаштування параметрів, а також статистики і аналізу. Агроном повинен стримати можливість конфігурувати радники на основі числових та перелічуваних користувачьких параметрів.

Таким чином будуть збережені основні переваги створеної раніше системи обліку врожайності полів – а саме універсальність системи для агронома. Агроном буде мати можливість налаштувати облік та аналіз під потреби свого господарства.

Аналіз має будуватись на основі радників. Радник – це налаштування аналізу, яке конфігурується агрономом чи надається розробником системи, створюється для визначеної культури та відносно обраного параметра, і встановлює допустимі значення цього параметра відносно культури радника.

Радник може базуватись на основі числового чи перелічуваного параметра. У разі числового параметра, користувач задає нижню границю параметру, оптимальне значення параметру, та верхню границю параметру. У разі перелічуваного параметру, агроном має задати числовий коефіцієнт відносно кожного значення перелічуваного параметру. Числовий коефіцієнт може приймати значення у межах  $[0;1]$ .

Так як перелічувані типи є ключовими для аналізу таких показників як попередник чи тип ґрунту, було вирішено додати цей тип до системи обліку врожайності полів для подальшого використання.

Радник числового типу містить наступну інформацію:

- Назва раднику
- Сільськогосподарська культура радника
- Числовий параметр системи який аналізується радником
- Нижнє граничне значення параметру
- Оптимальне значення параметру
- Верхнє граничне значення параметру

Радник перелічуваного типу містить наступну інформацію:

- Назва раднику
- Сільськогосподарська культура радника
- Перелічуваний параметр системи який аналізується радником

• Матив відношень між значеннями перелічуваного типу: значення типу –  
коєфіцієнт

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

## РОЗДІЛ 2 МОДЕЛЮВАННЯ СИСТЕМИ

# НУБІП України

### 2.1 Варіанти використання

# НУБІП України

У системі обліку врожайності полів існує чотири ролі: агронома, керівника, спеціаліста та адміністратора. Функціонал аналізу та статистики найбільш відповідає ролям агронома та керівника, так як вони є відповідальними за аналіз та прийняття рішень щодо планування сівозмін.

# НУБІП України

Наступні зміни були зроблені щодо ролей користувачів системи:

- Агроном

# НУБІП України

При встановленні модулів аналізу та статистики, роль агронома отримує найбільше додаткового функціоналу. Агроном є відповідальним за планування сівозмін, тому має доступ до аналізу культур, пошуку найбільш оптимальної культури для поля, перегляду усієї статистики. Також, агроном є відповідальним за створення і конфігурацію радників у системі.

# НУБІП України

- Керівник

# НУБІП України

Керівник не є відповідальним за конфігурацію радників чи безпосереднє прийняття рішення щодо планування сівозмін. Але, керівник може використовувати статистику для аналізу фінансових ефектів вирощування сільськогосподарських культур. Керівник є відповідальним за затвердження плану сівозмін, що є розробленим агрономом. Додатково керівник має доступ на читання радників та аналіз сівозмін, як і до інших можливостей системи.

# НУБІП України

• Спеціаліст

НУБІП України

Роль спеціаліста не отримує додаткових можливостей з доданням модулів аналізу та статистики. У контексті статистики і аналізу спеціаліст відповідає за ведення історії полів, тобто завдяки ролі спеціаліста система отримує дані для подальшого аналізу і функціонування системи обліку врожайності полів.

• Адміністратор

НУБІП України

Роль адміністратора не змінюється при доданні модулю аналізу і статистики.

Основною задачею адміністратора є управління обліковими записами за адміністративними розпорядженнями керівника агрофірми та підтримка системи у робочому стані.

НУБІП України

Описані вище варіанти використання можна зобразити на діаграмі прецедентів (рис. 4). Частина варіантів використання яка відноситься до модулю обліку врожайності полів була опущена для спрощення діаграми.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

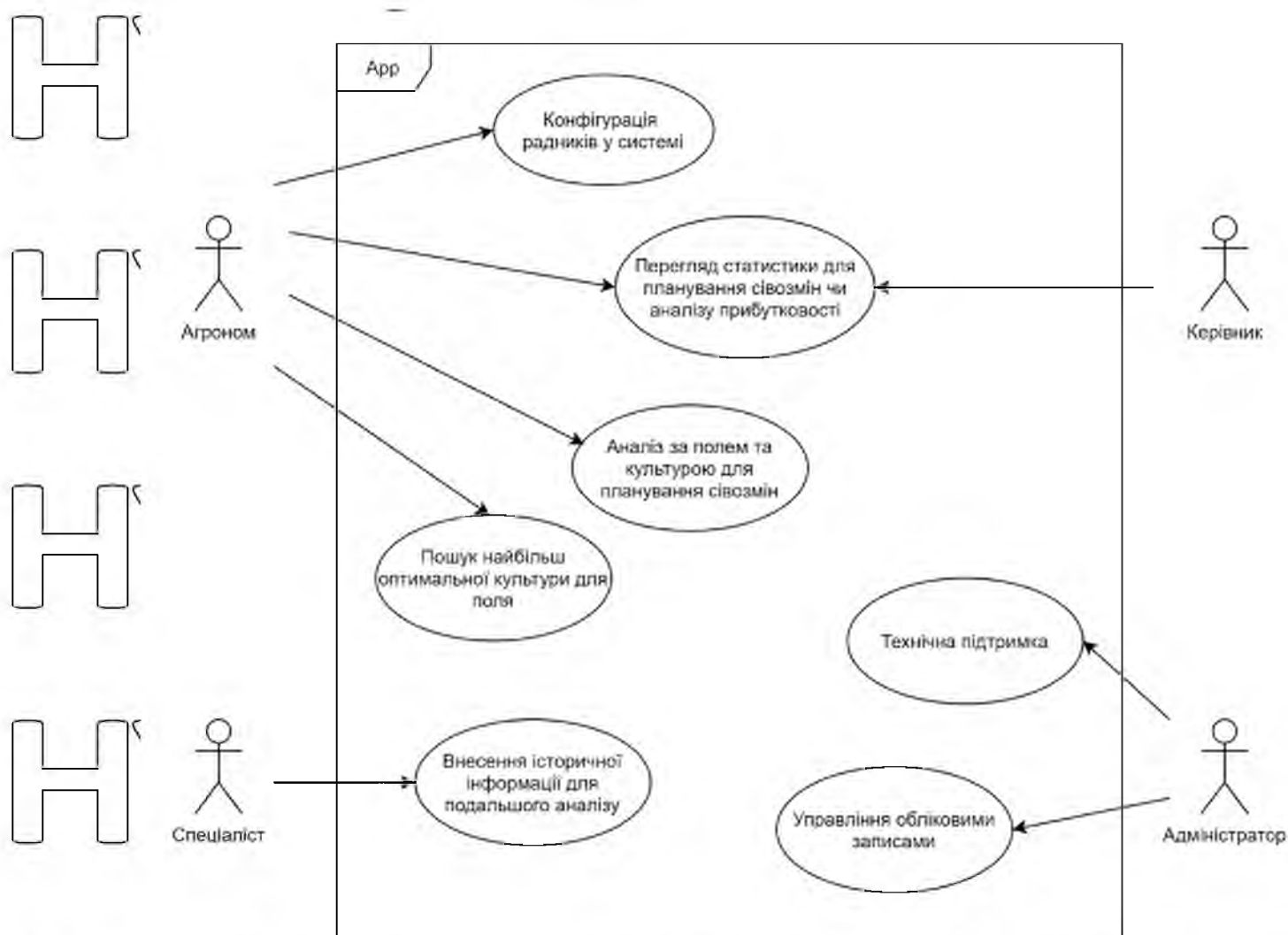


Рис. 4 Діаграма випадків використання

Для роботи системи адміністратор, за розпорядженням керівника, має створити облікові записи для співробітників підприємства.

## 2.2 Конфігурація радників

Для підвищення ефективності аналізу відносно окремої агрофірми, агроном підприємства має можливість конфігурувати радників у системі.

Радники також дозволяють додати функціонал аналізу до новостворених параметрів агрофірми.

Для створення радника, агроном спочатку має обрати назву та параметр, який буде аналізуватися радником. Цей параметр може мати числовий чи перелічуваний тип. Числовий тип потребує введення оптимального показника параметру, а також нижню та верхню границю. Чим більше відхилення в сторону від оптимального показника, і чим ближче дійсне значення до верхньої чи нижньої границі, тим менший коефіцієнт відповідності параметра. У випадку перелічуваного типу, для кожного відношення задається окремий коефіцієнт, який повертається радником у разі співпадання значення параметру.

Отримані коефіцієнти завжди мають однакову вагу в процесі пошуку оптимальної культури для поля. У разі, якщо під час аналізу відбувається ситуація, при якій недостатньо даних для формування поради, радник не виконується і його коефіцієнт не враховується в агрегованому результаті чи в підборі оптимальної культури. Така ситуація може відбуватись при доданні нового параметру у системи, бо більш старі історичні записи не будуть мати інформації по новому параметру. В такому разі буде доречним додати нові історичні записи до кожного поля, що є відповідальністю спеціаліста.

За потреби, на один параметр можна створити декілька радників, у тому числі одного й того самого типу.

Описаний вище процес конфігурування радників можна коротко зобразити на діаграмі активності (рис. 5).

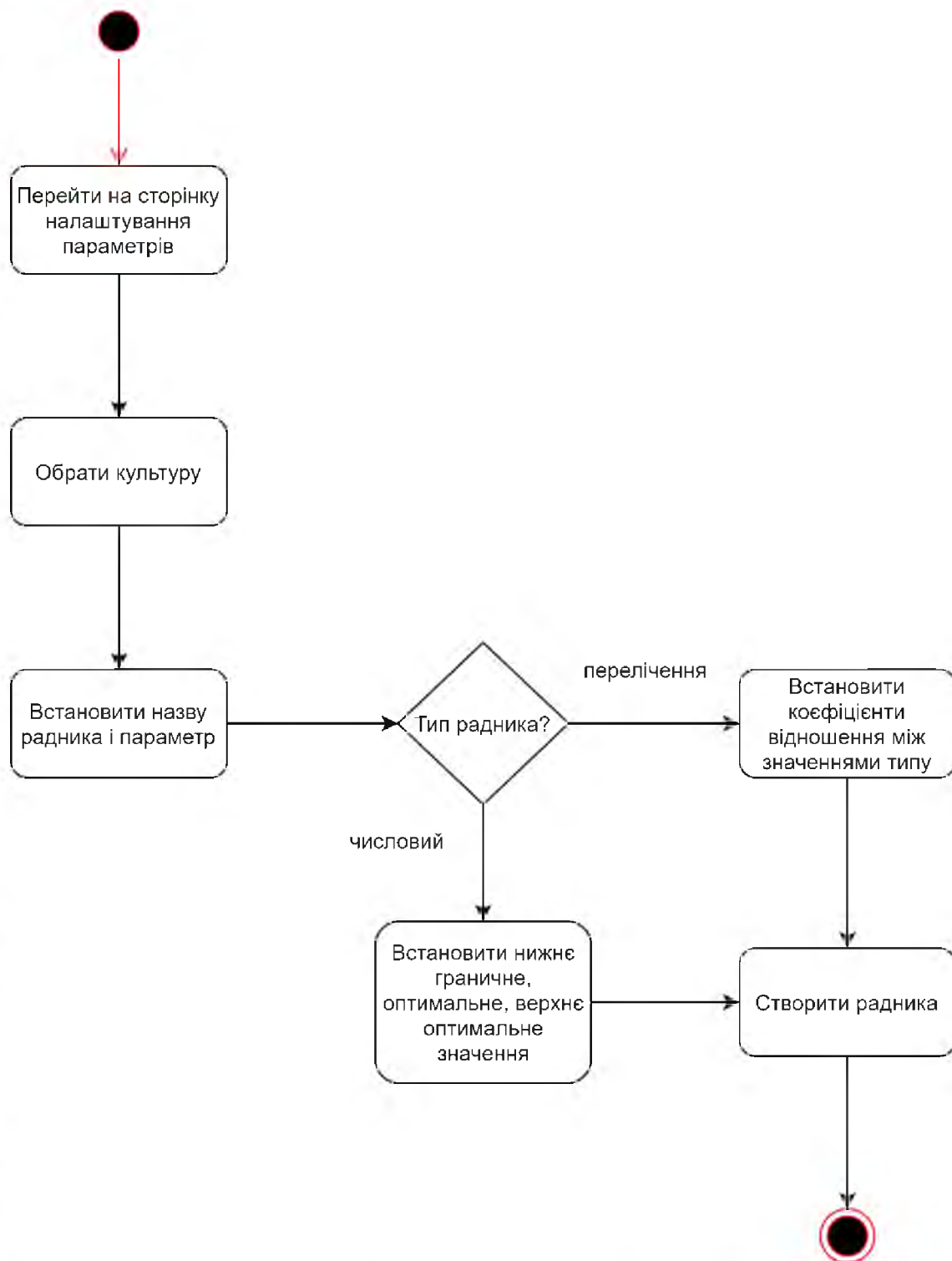


Рис. 5 Діаграма активності створення радника



# НУБІП УКРАЇНИ

## 2.3 Підбір оптимальної культури для поля

Одним з основних варіантів використання запропонованої системи є підбір оптимальної культури для окремого поля. Множина створених попередньо радників дозволяє отримати об'єктивну оцінку відповідності сільськогосподарської культури до поля. Для цього необхідно агрегувати результати аналізу усіх радників культури для отримання єдиної оцінки відповідності.

Представимо типовий варіант використання даного функціоналу. Агроному необхідно спланувати сівозміну. Для цього агроном може перейти на сторінку аналізу сівозміни та обрати поле для аналізування. Далі, агроном може викликати підбір найбільш оптимальної культури для поля. Після обробки запиту, він отримає результат аналізу за усіма культурами зареєстрованими у системі у відсортованому вигляді. Перший рядок таблиці і буде найбільш відповідною культурою до поля.

Після знаходження найбільш оптимальної культури, агроном має можливість ще більше оптимізувати процеси вирощування, викликавши більш детальний аналіз за конкретною (найбільш оптимальною) культурою. Це надає агроному результати аналізу за кожним окремим радником. На практиці це може надати агроному наступні переваги. Наприклад, агроном може побачити, що обране поле є у цілому сприятливим для вирощування культури, але ситуація зі вмістом азоту у ґрунті не є цілком оптимальною. За рахунок цієї інформації агроном може прийняти рішення об оптимізації подальшого вирощування – наприклад, вирішити застосувати азотні добрива перед початком вирощування культури.

Описаний приклад зображено у вигляді діаграми активності на рис 6.

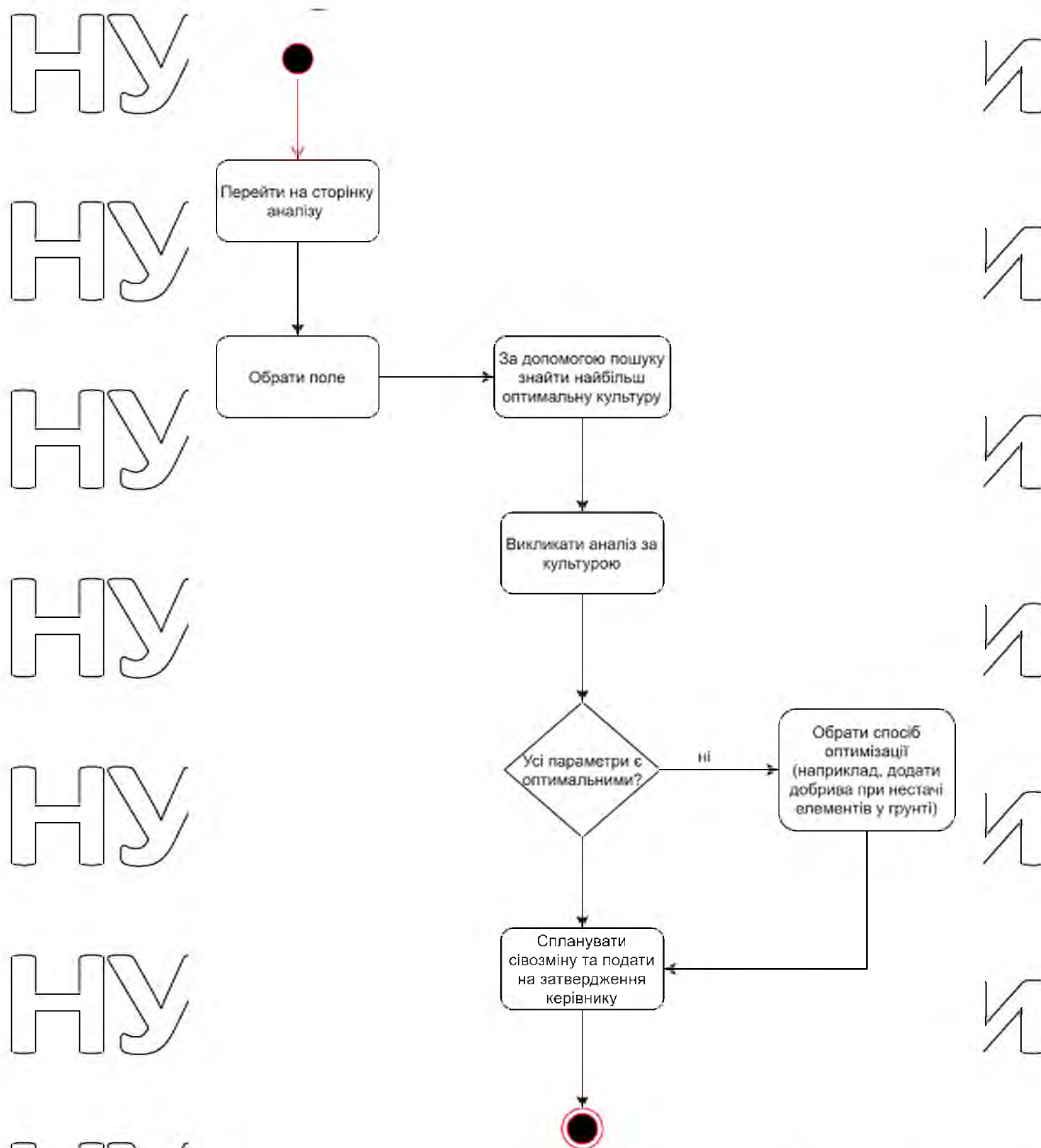


Рис. 6 Діаграма активності підбору оптимальної культури

## 2.4 Використання статистичного функціоналу

Статистичний функціонал може використовуватись як для аналізу динаміки стану поля за часом, так і для статистичного аналізу з урахування культури. Такий аналіз дозволяє простежувати як кожна культура діє на стан ґрунту поля, що дозволяє покращити планування сівозмін.

У випадку старих полів з великою кількістю історичних даних, відображення графіків може бути незручним для аналізу. Для таких ситуацій необхідно додати до системи можливість обмежити часовий проміжок, за яким необхідно будувати графік.

Загальний алгоритм планування сівозміни з використанням статистичної інформації можна представити так: агроном чи керівник переходить на сторінку статистики додатку, обирає необхідне поле для планування, встановлює отримання статистики за усіма культурами та отримує дані за увесь проміжок часу. Якщо кількість даних ускладнює аналіз, то агроном може встановити часові обмеження, отримує дані і аналізує динаміку стану поля.

Наступним кроком в типовому алгоритмі планування є аналіз за часом, полем та культурою. Агроном обирає попередню культуру на сторінці статистики та завантажує нові дані. Отримавши дані, агроном може прослідкувати як культура впливає на стан поля, і використати ці дані для планування сівозміни. Такий функціонал дозволяє планування на упередження, тобто на декілька кроків вперед, що є важливим у довгострокових сівозмінах.

Даний алгоритм візуалізовано на діаграмі активності на рис. 7

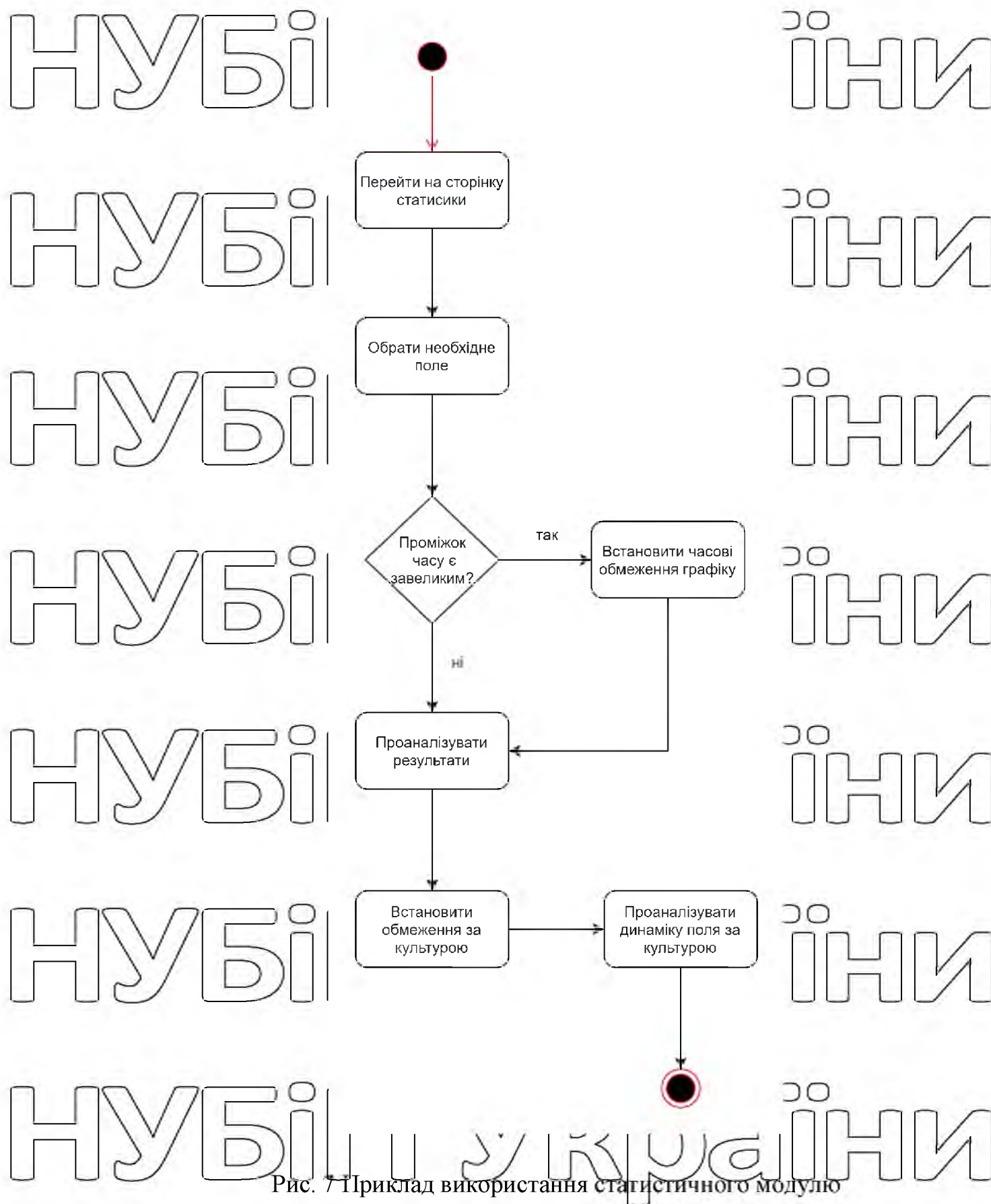


Рис. 7 Приклад використання статистичного модулю

# НУБІП України

## РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ

### 3.1 Розробка логічної моделі даних

Гнучка система радників, яка дозволяє конфігурувати налаштування аналізу для користувальницьких параметрів потребує використання реляційної бази даних.

Першим кроком проектування реляційної бази даних є побудова логічної моделі даних. Логічна модель даних, або логічна схема — модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання [6]. Реляційну логічну модель бази даних було вирішено побудувати за допомогою програмного забезпечення під назвою CA ERwin Data Modeler.

Реляційна база даних це база даних, побудована на основі реляційної моделі, тобто БД, що має табличний спосіб вистави даних, а на зовнішньому рівні, що задається набором однорідних таблиць. Кожний об'єкт записується рядком у таблиці. Рядок називається записом. Запис складається з полів різного типу [7].

У якості основи була використана модель бази даних системи обліку врожайності сільськогосподарських полів агрофірми. До існуючої моделі були додані наступні таблиці:

- Таблиця ‘field attribute enum’ відповідає за доступні значення перелічуваних типів, та має відношення «багато до одного» до таблиці ‘field attribute’ (це відношення означає що значення належить визначеному атрибуту)

- Таблиця ‘field advisor numeric’ відповідає за конфігурацію числового радника підсистеми аналізу. Таблиця має два відношення: відношення «багато до одного» відносно таблиці ‘field attribute’, що позначає

значення якого атрибуту аналізує цей радник. Інше відношення, типу “багато до одного” до таблиці ‘field\_attribute\_enum’ позначає відносно якого атрибуту працює радник. З функціональної точки зору, на даний

момент радники завжди працюють відносно окремої культури,

ідентифікатор якої буде зазначений у відношенні радника, але в моделі бази даних такого обмеження немає. Такий підхід було обрано для подальшого розвитку системи (додання радників які працюють відносно інших параметрів, наприклад ґрунту).

- Таблиця ‘field\_advisor\_enum’ відповідає за конфігурацію радників перелічуваного типу. Таблиця копіює відношення таблиці ‘field\_advisor\_numeric’.

- Таблиця ‘field\_advisor\_enum\_relation’ відповідає за відношення між значеннями перелічуваного типу та їх відповідними коефіцієнтами. Таблиця має два відношення: відношення типу “багато до одного” відносно таблиці ‘field\_advisor\_enum’, що показує якому раднику належить це функціональне відношення, а також відношення типу ‘багато

до одного’ до таблиці ‘field\_attribute\_enum’, яке показує до якого значення перелічуваного типу належить коефіцієнт.

Описану вище модель було декомпозовано до третьої нормальної форми.

Базове відношення знаходиться в третій нормальній формі відносно множини функціональних залежностей тоді і тільки тоді, коли воно знаходиться у другій нормальній формі і кожен неключовий атрибут повністю функціонально залежить від ключів [8].

Можна аргументувати, що описана модель знаходиться в третій нормальній формі, бо:

- Всі атрибути моделі мають лише скалярні значення.

# НУБІП України

- Кожен неключовий атрибут залежить від первинного ключа.
- Кожен неключовий атрибут є нетранзитивно залежним від первинного ключа.

Візуалізація логічної моделі бази даних наведена на рис. 8.

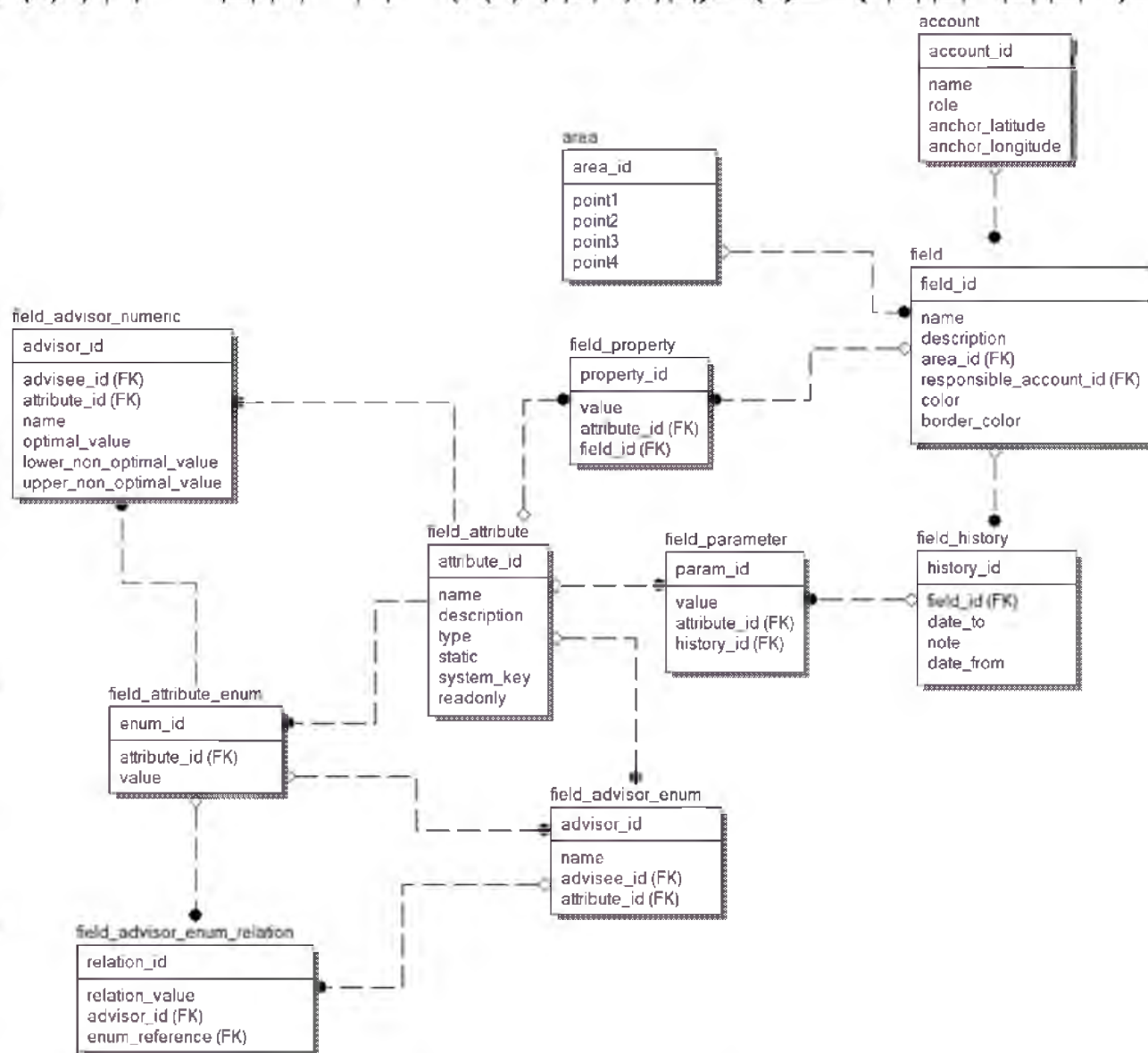


Рис 8. Логічна модель бази даних

### 3.2 Вибір системи візуалізації діаграм

Для побудови візуалізації даних, у тому числі часових графіків, доречно використовувати існуючі рішення.

До вибору системи візуалізації було поставлено наступні вимоги:

- Система повинна інтегруватись з клієнтським фреймворком Angular 9, на основі якого працює решта клієнтської частини додатку.
- Система повинна підтримувати часові графіки.
- Система повинна мати як найбільш широкую підтримку інших видів візуалізації, для спрощення розробки системи у майбутньому.
- Рішення повинне працювати швидко навіть при великій кількості часових точок на графіку.
- Рішення має мати новітній інтерфейс, який можна модифікувати.
- Рішення має мати відкриту ліцензію.

Керуючись цими вимогами, було вирішено використовувати систему ngx-charts для візуалізації даних у додатку.

NGX-charts — це декларативний фреймворк для побудови діаграм з відкритим кодом для angular2+. Його розробляє компанія Swimlane. Він використовує Angular для візуалізації та анімації елементів SVG з усіма його перевагами типу зв'язування, швидкодії та використовує d3 для математичних функцій, масштабування, побудови генераторів форм, осей тощо [9].

Для візуалізації були обрані діаграми з областями.

Діаграми з областями поєднують лінійну та стовпчасту діаграму, щоб показати, як змінюються числові значення однієї чи кількох груп із прогресом



другої змінної, як правило, у часі. Діаграми з областями відрізняються від лінійних діаграм додаванням затінення між лініями та базовою лінією, як у гістограмі [10].

### 3.3 Розробка операції підбору оптимальної культури

Операція пошуку в системі є однією з ключових у системі. Процес пошуку починається з моменту, коли користувач, обравши необхідне поле у формі, нажимає на кнопку підбору оптимальної культури. Після цього, веб частина додатку відправляє запит на підбір оптимальної культури на веб-сервер. Веб-сервер, у свою чергу викликає контролер фреймворку Spring, який в свою чергу передає управління контролеру який відповідає за аналіз у системі. Метод контролеру перевіряє отримані дані і викликає сервіс аналізу і підбору.

Сервіс аналізу спочатку завантажує поле і усі зареєстровані культури. Після цього, для кожної культури викликається аналіз відповідності.

Аналіз відповідності для окремої культури починається із завантаження списку радників із бази даних. Для кожного радника сервіс запускає логіку перевірки відповідності параметра. Спочатку відбувається перевірка атрибуту. Якщо атрибут радника є статичним, то сервіс обирає поточне значення статичного атрибуту. Якщо атрибут не є статичним, то сервіс завантажує з бази даних останній історичний запис, і використовує значення параметру з цього запису. Далі виконується логіка специфічна до типу радника.

У випадку радника числового типу, сервіс аналізу вираховує ступінь відхилення поточного значення від оптимального значення встановленого у раднику. Результат роботи перевірки є коефіцієнт відповідності у межах  $[0; 1]$ , де 0 означає повну невідповідність (у разі якщо поточне значення є нижчим за нижнє

неоптимальне, чи вищим за верхнє неоптимальне значення встановлене у раднику), а 1 – повну відповідність (коли поточне значення і оптимальне значення є рівними).

У випадку радника перелічуваного типу, сервіс завантажує список залежностей для радника, і шукає потрібну залежність за ідентифікатором перелічуваного типу. Якщо залежність була знайдена, результатом роботи стає коефіцієнт що є вказаним у значенні залежності. Якщо потрібної залежності не існує, то результатом є null, що означає неможливість виконання радника.

Після обробки усіх радників, підраховується агрегований результат як середнє значення усіх коефіцієнтів. Якщо є радники, які не обробились через відсутність залежності чи інформації, то вони не включаються у звіт та агрегований результат.

Після обробки кожної культури, сервіс аналізу сортує результати за агрегованим коефіцієнтом у порядку зменшення, і повертає результат на клієнтську частину додатку у вигляді звіту.

Отримавши звіт, клієнтська частина завантажує дані звіту у таблицю і відображає користувачу додатка, дозволяючи йому провести подальший аналіз та планування сівозмін.

Описану логіку можна спрощено представити на діаграмі послідовності (рис. 9).

НУБІП України

НУБІП України

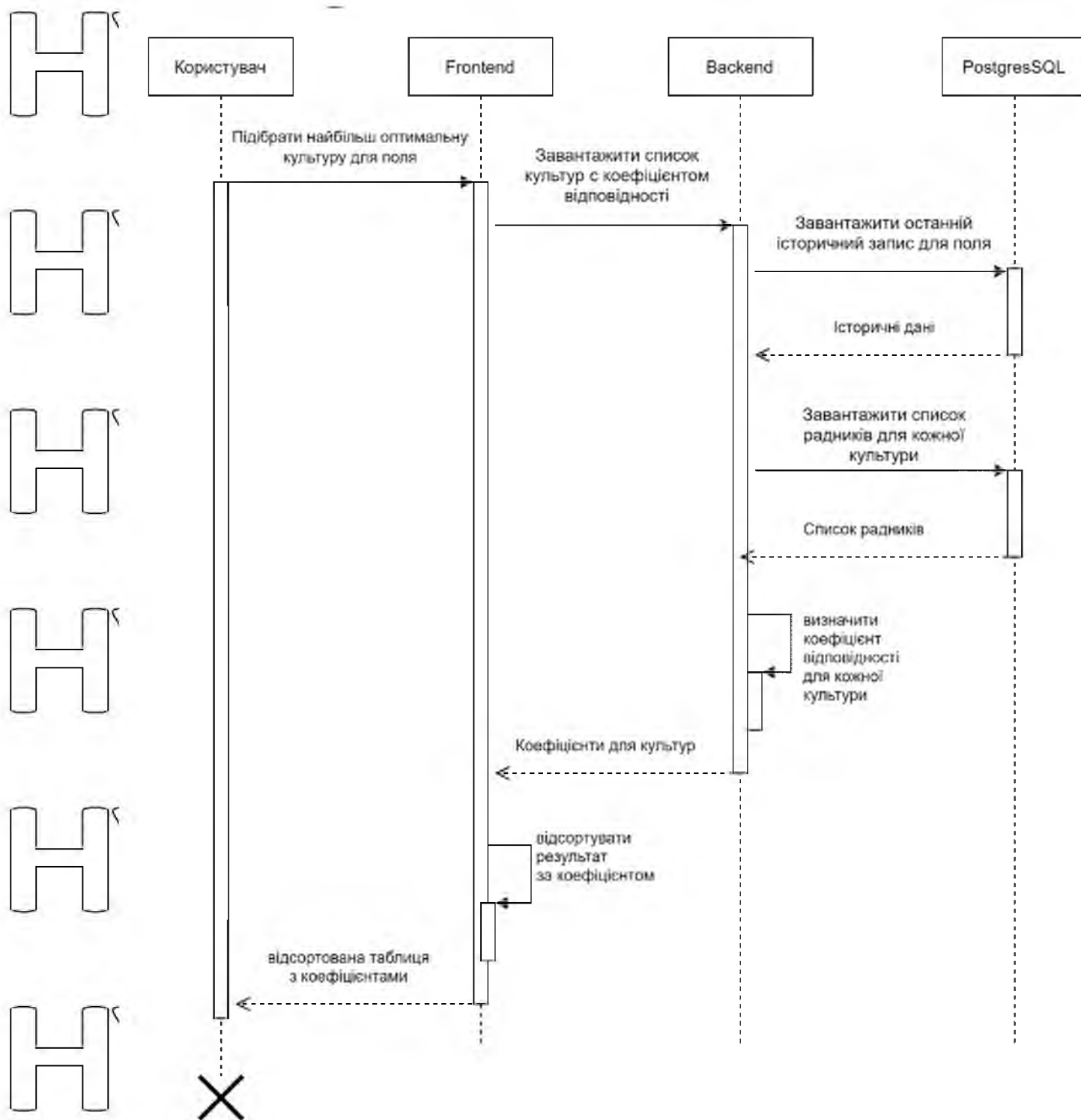


Рис. 9 Діаграма послідовності підбору культур для поля

НУБІП України

### 3.4 Розробка операції завантаження статистики поля

Операція завантаження статистики є однією ключовою операцією у системі. Процес завантаження статистики починається з переходу користувача на сторінку статистики. Обравши необхідне поле і, опціонально, культуру, користувач має натиснути на кнопку отримання статистики. Після натиснення на кнопку, клієнтська частина додатку відправляє запит на веб-сервер, а той передає управління фреймворку Spring, а той в свою чергу викликає контролер статистики.

Отримавши дані, контролер статистики викликає сервіс статистики. Сервіс статистики завантажує історичні дані для поля в бази даних, та форматує їх до наступного формату типу JSON (рис.10).

```

1 [
2   {
3     "name": "Назва графіку",
4     "series": [
5       {
6         "name": "2010-01-01",
7         "value": 7300000
8       },
9       {
10        "name": "2011-01-01",
11        "value": 8940000
12      }
13    ]
14  },

```

Рис 10. Формат статистичних даних

JSON — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією) [11].

Для кожного не статичного параметру в системі будується свій графік. У разі, якщо в системі є менше двох записів з окремим параметром, то такий графік не будується і не відображається користувачу системи.

Дані у зображеному вище форматі передаються на клієнтську частину додатку, де викликається бібліотека ngx-charts, яка будує відображення графіку у форматі SVG.

SVG є мовою яка походить із XML, для опису двомірної векторної графіки.

SVG по суті є графікою, так же, як XHTML — текстом. SVG за своїми можливостями наближається до запатентованої технології Adobe Flash, але відрізняється від неї тим, що є SVG є рекомендацією W3C, тобто стандартом, і тим, що це формат, заснований на XML, на відміну від закритого формату Flash. Він явно спроектований для роботи з іншими стандартами W3C, такими, як CSS, DOM (en-US) і SMIL [12].

Описаний вище процес можна представити на діаграмі послідовності (див рис. 11).

### 3.5 Архітектура серверної частини додатку

Серверна частина модулів аналізу та статистики написана на мові Java 11.

Java є мовою програмування, за допомогою якої розробники програмного забезпечення можуть створювати різні прикладні додатки для комп'ютерів, смартфонів, планшетів та інших інтелектуальних пристроїв. Особливістю програми на Java є те, що вони можуть запускатись на будь-яких комп'ютеризованих пристроях, які працюють під різними операційними системами, причому без повторної компіляції коду [13].

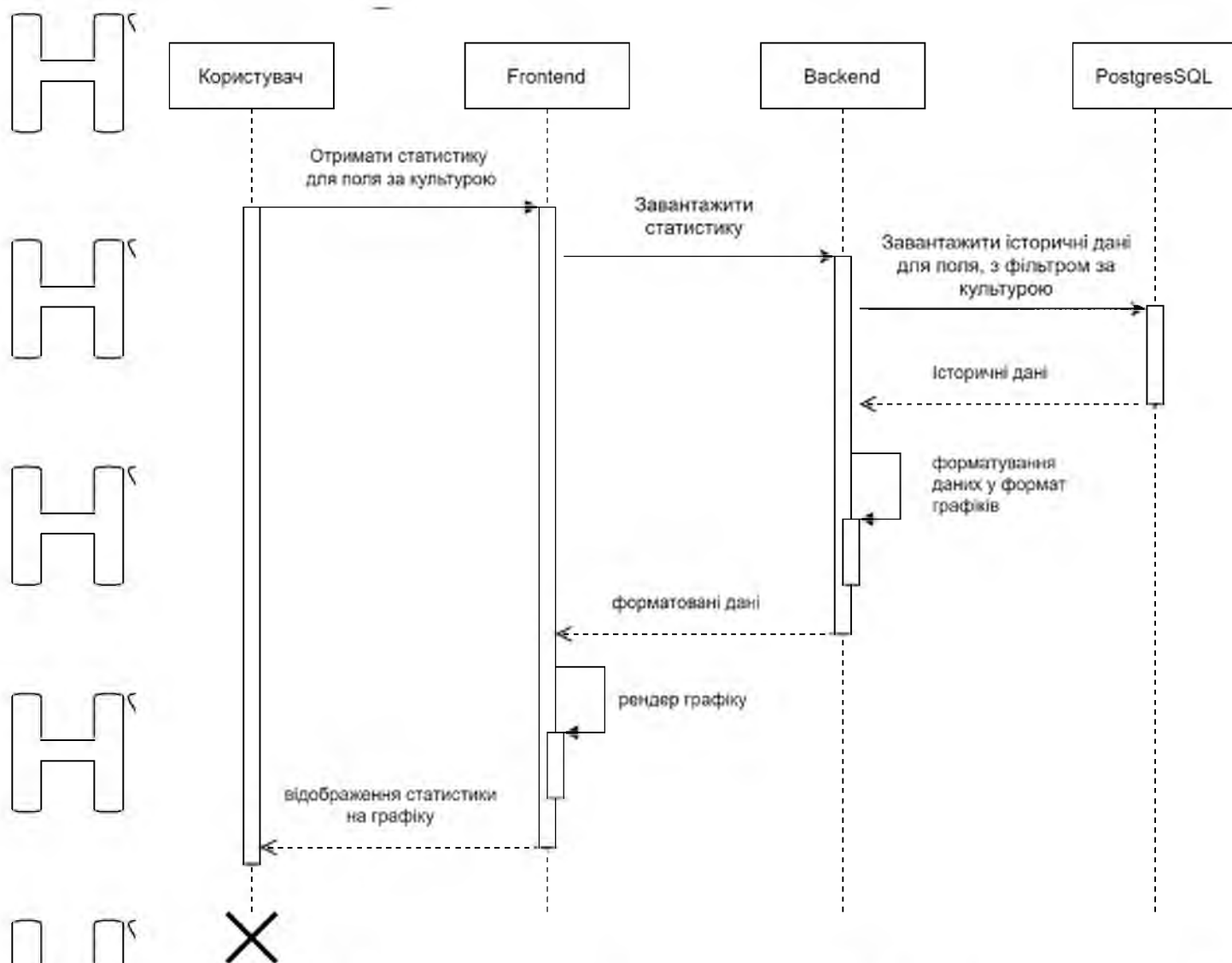


Рис. 11 Діаграма послідовності отримання статистики

Мова програмування Java була обрана через її основні переваги:

- Статична типізація
- Автоматичне керування пам'яттю
- Кросплатформеність
- Велика кількість бібліотек
- Керування залежностями за допомогою Maven
- Гарна підтримка у сфері бекенд розробки

У якості основного MVC-фреймворку було обрано **Spring**.

По суті, Spring пропонує контейнер, який часто називають контекстом програми Spring, який створює та керує компонентами програми. Ці компоненти, або біни, зв'язуються разом у контексті програми, і утворюють таким чином, завершене програмне забезпечення, як цегла, розчин, брус, цвяхи, сантехніка та електропроводка зв'язані між собою створюють будинок [14].

Для розробки було обрано метод конфігурування Spring на основі анотацій, як більш зручний та гнучкий спосіб розробки.

Spring складається з великої кількості модулів, які можна підключати до проекту у разі необхідності. Наступні модулі були використані для написання серверної частини додатку:

- Spring Framework – головний модуль фреймворку, який реалізує патерн ін'єкції залежностей і керує життєвим циклом усіх компонентів системи.
- Spring Security – модуль забезпечення аутентифікації і авторизації користувачів у системі.
- Spring Boot – модуль автоконфігурування системи, який звільнює від необхідності вручну конфігурувати більшість інших модулів Spring та частину зовнішніх залежностей.
- Spring Data – модуль, що дає можливість автоматизувати роботу з БД завдяки автоматичному створенню і реалізації класів-репозиторіїв.
- Spring MVC – частина Spring, що дозволяє будувати API для звернення з фронтенду додатку.

Для спрощення роботи з базою даних було застосовано модуль **Hibernate**.

НУБІП УКРАЇНИ  
 Hibernate – засіб відображення між об'єктами та реляційними структурами (object-relational mapping, ORM) для платформи Java. Hibernate є вільним програмним забезпеченням, яке поширюється на умовах GNU Lesser General Public

License. Hibernate надає легкий для використання каркас (фреймворк) для відображення між об'єктно-орієнтованою моделлю даних і традиційною реляційною базою даних [15].

НУБІП УКРАЇНИ  
 Для автоматичної генерації типових конструкцій мови Java було використано бібліотеку Project Lombok. Він дозволяє автоматично генерувати геттери, сеттери, білдери об'єктів та інші типові конструкції та патерни.

НУБІП УКРАЇНИ  
 Веб-сервером додатку виступає **Apache Tomcat**

НУБІП УКРАЇНИ  
 Програмне забезпечення Apache Tomcat - це реалізація з відкритим кодом технологій Java Servlet, JavaServer Pages, Java Expression Language та Java WebSocket [16].

НУБІП УКРАЇНИ  
 Для збірки та побудови додатку було обрано технологію **Apache Maven**.

НУБІП УКРАЇНИ  
 Apache Maven - це програмне забезпечення для управління та збірки проектів. Він заснований на концепції об'єктної моделі проекту (POM). Maven може керувати збіркою проекту, складенням звітів і документацією з центральної частини інформації [17]. Використовуючи Apache Maven можна збирати весь проект у один .jar файл для простого розгортання додатку на будь-якій системі, що підтримує JVM.

НУБІП УКРАЇНИ  
 Процес розробки додатку відбувається з використанням середовища розробки ПЗ **IntelliJ IDEA**.

НУБІП УКРАЇНИ  
 Програмне забезпечення JetBrains IntelliJ IDEA – це провідне середовище швидкої розробки на мові Java, включаючи Java 8. IntelliJ IDEA являє собою високотехнологічний комплекс тісно інтегрованих інструментів програмування, що



включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, потужні інструменти рефакторинга коду, вбудовану підтримку технологій J2EE, механізми інтеграції з середовищем тестування Ant / JUnit і

системами управління версіями, унікальний інструмент оптимізації та перевірки коду Code Inspection, а також інноваційний візуальний конструктор графічних інтерфейсів [18].

Сукупність залежностей серверної частини додатку представлено на діаграмі компонентів (рис. 12).

### 3.6 Архітектура клієнтської частини додатку

У рамках кваліфікаційної роботи було розроблено веб-інтерфейс для додатку.

Веб-інтерфейс є найбільш універсальним рішенням, тому що його можна використовувати на будь-яких пристроях з підтримкою браузера: комп'ютерах, смартфонах, планшетах, розумних годинниках тощо. Але веб-інтерфейс не є

жорстко прив'язаним до додатку – у разі необхідності залишається можливість побудувати, наприклад, додаток для смартфонів на Android чи iOS, використовуючи API серверної частини додатку.

Для побудови верстки веб-клієнту було використано **Bootstrap**.

Bootstrap – найпопулярніший CSS фреймворк від компанії Twitter для адаптивної верстки сайтів. Bootstrap – це набір готових CSS файлів, який можна підключити до сайту, щоб швидко і якісно зробити його «мобільним» – тобто

таким, щоб він коректно відображався на телефонах і планшетах. Основа Bootstrap – модульна сітка з 12 колонок, яка вміє перелаштовуватися в залежності від розміру екрану пристрою відвідувача сайту [19].

# НУБІП УКРАЇНИ

Bootstrap дозволяє швидко розробляти інтерфейс для клієнтської частини додатку, який масштабується в залежності від розміру екрану користувача

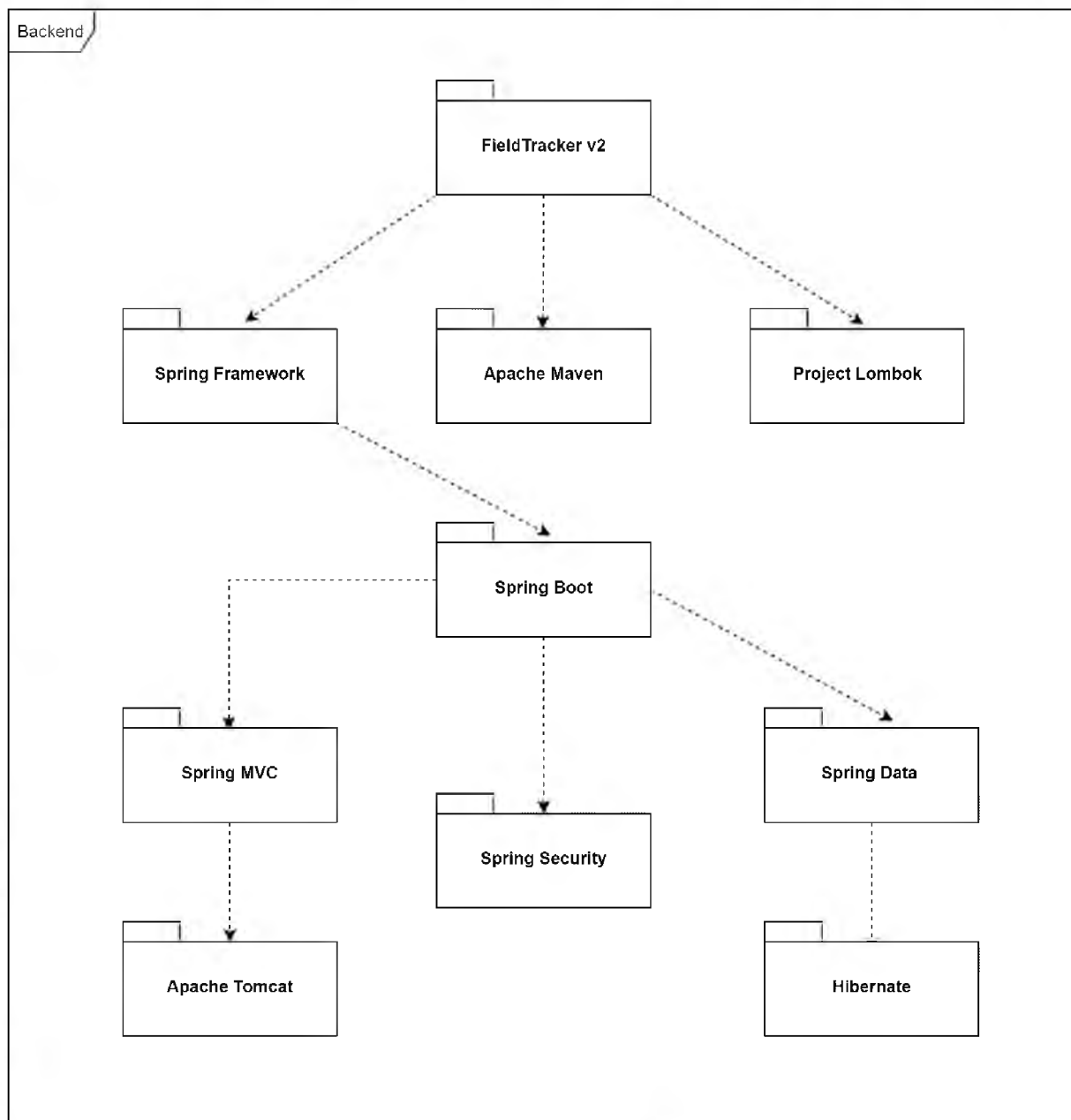


Рис 12. Діаграма компонентів серверної частини додатку

# НУБІП УКРАЇНИ

У якості основної мови для розробки веб-клієнту використовується TypeScript. TypeScript це компільована до JavaScript мова програмування, яка додає

концепції об'єктно-орієнтованого програмування та статичну типізацію до звичного JavaScript [20]. Використання статично типізованої мови дозволяє будувати великі за обсягом системи зі значно меншим ростом складності при збільшенні кодової бази.

Основним веб-фреймворком додатку було обрано **Angular 9**. Angular являє собою фреймворк від компанії Google для створення клієнтських додатків. Він націлений на розробку SPA-рішень (Single Page Application), тобто односторінкових додатків. В цьому плані Angular є спадкоємцем іншого фреймворка AngularJS. У цей же час Angular це не нова версія AngularJS, а принципово новий фреймворк. Angular надає таку функціональність, як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделей в іншому, шаблонах, маршрутизації і так далі [21]. Основною перевагою фреймворку Angular для додатку є підтримка шаблонізації, через велику кількість динамічного контенту, яка впливає із характеру додатку. Можливість інтегрування з іншими рішеннями як Google Maps також є корисною під час розробки проекту.

Angular автоматизує такі частини додатку як авторизація, безпека, шаблонізація, рутинні, захист від атак, інтеграція з іншими рішеннями тощо.

Для побудови візуалізації статистики у вигляді лінійних графіків було використано модуль для Angular під назвою **ngx-charts**.

Ngx-charts дозволяє автоматизовано будувати візуалізацію графіків в додатку Angular. Цей модуль має широку підтримку з боку opensource програмістів та підтримує велику кількість різних графіків та візуалізацій. Даний модуль також дозволяє кастомізувати відображення графіків, роблячи їх більш доречними до тематики розроблюваного додатку.

Сукупність залежностей веб-клієнту використаних у додатку візуалізовано на діаграмі компонентів (рис. 13).

# НУБІП Україна

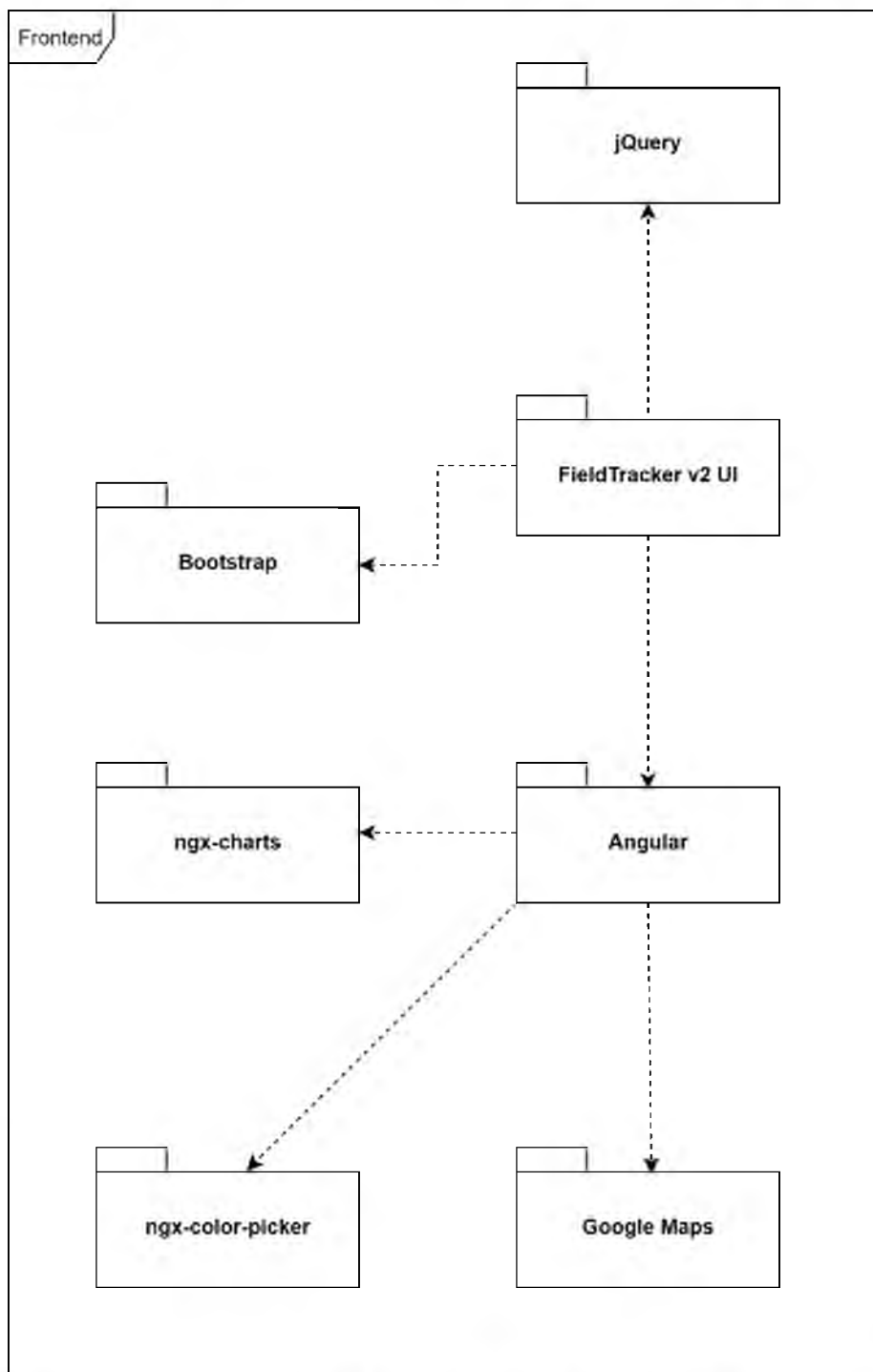


Рис. 13 Діаграма компонентів веб-клієнту додатку

## РОЗДІЛ 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 4.1 Отримання статистики полів

В процесі виконання проекту було розроблено веб-сторінку статистики полів. Після авторизації у системі, користувач має можливість перейти на сторінку статистики полів (див рис. 14).

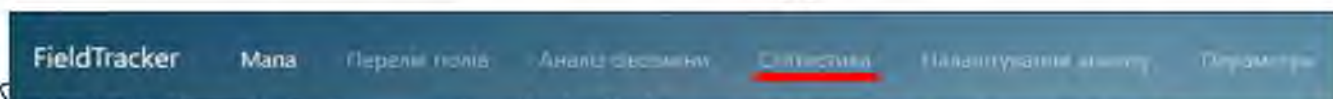


Рис. 14 Перехід на сторінку статистики

На сторінці користувач може встановити наступні значення:

- Поле
- Культура (для всіх культур, для окремої культури)
  - Від (дата чи її відсутність)
  - До (дата чи її відсутність)

Після встановлення значень, користувач має натиснути на кнопку завантаження статистики.

У разі, якщо у системі недостатньо даних для побудови навіть одного графіку за встановленими значеннями, система напише про помилку (рис. 15). Це зазвичай може відбутись у разі нового поля без історії, при аналізі за культурою яка вирощувалася менше двох разів на одному полі, або у разі відсутності даних за вказаним часовим проміжком.

Рис. 15 Помилка під час отримання статистики

У разі успішного завантаження, веб-клієнт відображає п'ять графіків, де п'ять – кількість нестатичних параметрів системи з більше ніж двома історичними значеннями.

Шкала  $x$  кожного графіку являє собою час, а шкала  $y$  – значення параметру у відповідний проміжок часу. Дата представлена в міжнародному форматі уууу-ММ-дд, наприклад – 2021-01-31. Назва графіку дорівнює назві параметру який представлений відповідним графіком. У кожному рядку може бути представлено два графіки, зліва направо, з необмеженою кількістю рядків на одній сторінці (див. рис. 16).

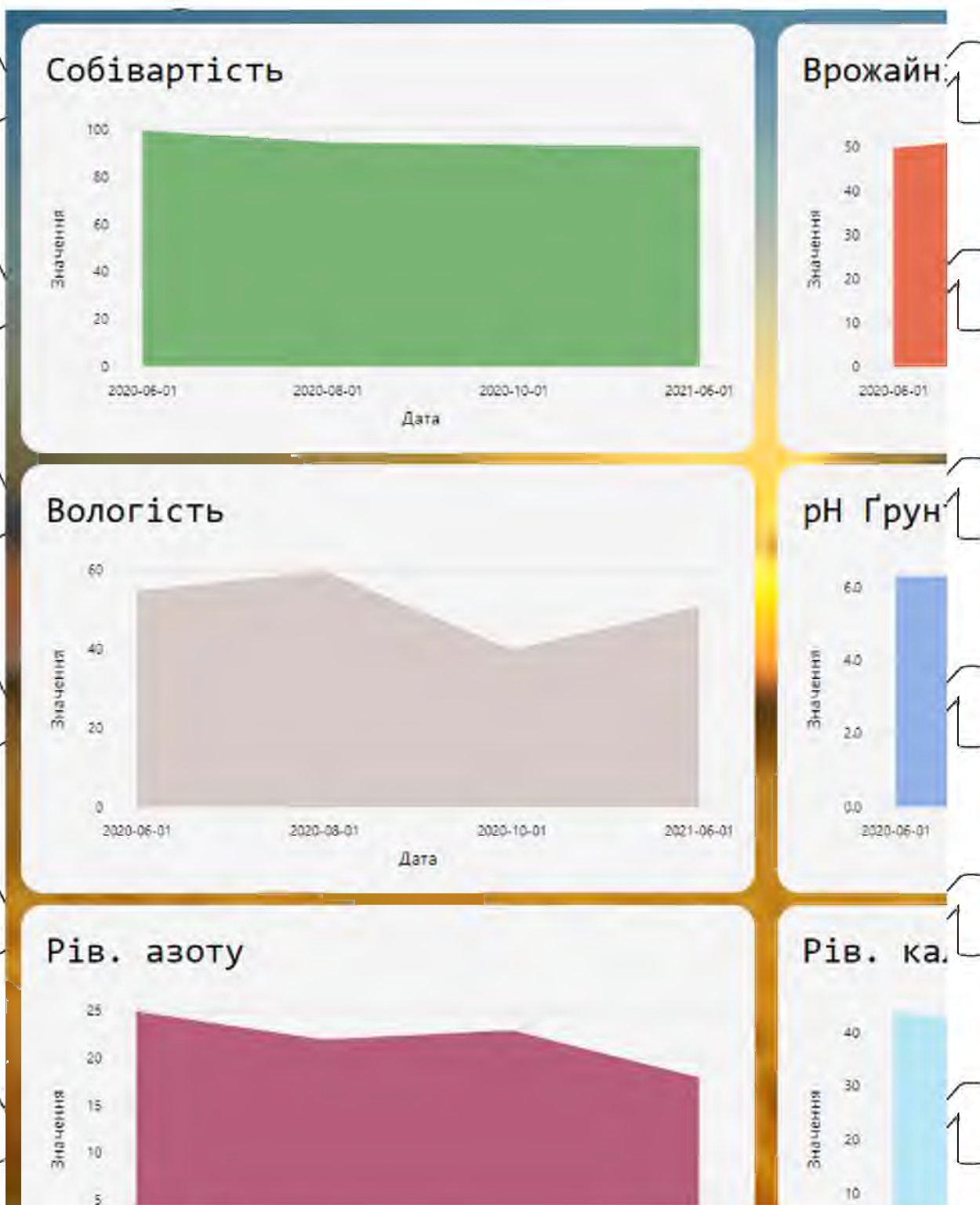


Рис. 16 Часові графіки за параметрами

## 4.2 Сторінка аналізу сівозміни

Перехід на сторінку аналізу сівозміни є доступним через панель навігації, після авторизації у системі з правами керівника чи агронома (див. рис. 17).



Рис. 17. Перехід на сторінку аналізу сівозмін

На сторінці аналізу сівозмін необхідно вказати параметри поля та культури. У списку представлені усі поля та культури що є зареєстрованими у системі, та кнопки аналізу за культурою та підбору оптимальної культури (див. рис. 18).

Рис. 18. Форма аналізу сівозміни

Після натиснення на кнопку аналізу за культурою, веб-клієнт виконає запит на сервер та отримає результати аналізу. Ці результати далі будуть відображені у таблиці з наступними стовпцями:

- Умова (назва радника)



• Для числових типів: значення у форматі “<значення параметра> при [<нижнє\_неоптимальне\_значення>...<оптимальне\_значення>...<верхнє\_неоптимальне\_значення>”

- Для перелічуваних типів: значення перелічуваного типу параметру що

аналізується радником, без відображення коефіцієнту відповідності

- Коефіцієнт відповідності (від 0 до 1)
- Підсумок (оптимальні умови, неоптимальні умови, не відповідає умовам)

У самому нижньому рядку відображається агреговане значення за усіма радниками та фінальний підсумок за полем і культурою.

Значення стовпців відповідності та підсумку змінюють свій колір в залежності від значення підсумку:

- Оптимальне значення – зелений колір
- Не оптимальне значення – жовтий колір
- Не відповідає умовам – червоний колір

Тобто, кольорове рішення повторює логіку світлофора.

У разі відсутності необхідних даних для радника, він не відображається у таблиці результатів аналізу за полем та культурою у системі. Цей варіант було обрано для зменшення візуальної завантаженості системи.

Приклад результату аналізу за культурою наведено на рис. 19.

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

| Результати аналізу         |                              |               |                     |
|----------------------------|------------------------------|---------------|---------------------|
| Умова                      | Значення                     | Відповідність | Підсумок            |
| рН Ґрунту для соняшника    | 6.45 при [6.00..6.40..6.80]  | 0.87          | Оптимальні умови    |
| Рів. азоту для соняшника   | 18 при [10.00..25.00..50.00] | 0.33          | Не оптимальні умови |
| Попередник для соняшнику   | Овес                         | 1             | Оптимальні умови    |
| Рів. фосфору для соняшника | 7.5 при [3.00..7.00..12.00]  | 0.9           | Оптимальні умови    |
| Рів. кальцію для соняшника | 21 при [9.00..22.00..34.00]  | 0.92          | Оптимальні умови    |
| <b>Агреговано</b>          | -                            | 0.844         | Оптимальні умови    |

Рис. 19 Результати аналізу за культурою

У разі натиснення кнопки підбору оптимальної культури, завантажується таблиця з деяко іншою інформацією, а саме:

- Назва аналізованої культури
- Коефіцієнт відповідності (від 0 до 1)
- Підсумок (оптимальні умови, неоптимальні умови, не відповідає умовам)

Значення стовпців відповідності та підсумку змінюють свій колір аналогічно функціоналу аналізу за культурою. Результати аналізу є відсортованими у порядку зменшення за коефіцієнтом відповідності культури.

Приклад таблиці результатів підбору оптимальної культури наведено на рис. 20.

Підбір оптимальної культури для поля

| Культура  | Відповідність | Підсумок             |
|-----------|---------------|----------------------|
| Соняшник  | 0.844         | Оптимальні умови     |
| Овес      | 0.585         | Не оптимальні умови  |
| Просо     | 0.811         | Не оптимальні умови  |
| Соя       | 0.426         | Не відповідає умовам |
| Кукурудза | 0.36          | Не відповідає умовам |
| Ячмінь    | 0.096         | Не відповідає умовам |

Рис. 20 Таблиця результатів підбору оптимальної культури

### 4.3 Налаштування аналізу

Для переходу на сторінку налаштування аналізу, необхідно авторизуватись користувачем типу агроном чи керівник, та перейти на сторінку налаштування аналізу на панелі навігації (рис.21).



Рис. 21 Перехід на сторінку налаштування аналізу

Сторінка налаштування аналізу складається з трьох логічних частин:

- Блок вибору культури
- Перелік радників на основі цифрових типів
- Перелік радників на основі перелічуваних типів

Блок вибору культури дозволяє завантажити радників, які відносяться до обраної культури. Він складається з одного поля культури та кнопка завантаження радників (див. рис. 22).

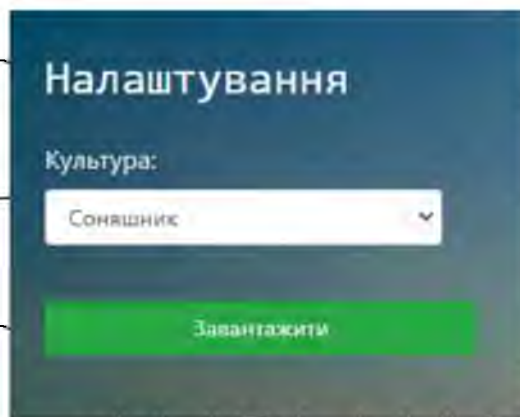


Рис. 22 Блок вибору культури

Наступним блоком є перелік радників на основі цифрових типів. Він складається з таблиці яка містить опис раднику, кнопки редагування та видалення раднику, та форму додання чи редагування раднику у вигляді нижнього ряду. Таблиця містить наступні рядки:

- Назва раднику
- Атрибут раднику
- Нижнє граничне значення
- Оптимальне значення
- Верхнє граничне значення
- Стовпець дій

Націнення на кнопку редагування раднику копіює поточні значення раднику в рядок редагування, та замінює кнопку створення на кнопку редагування.

Таблицю радників цифрового типу зображено на рис. 23.

| Назва                     | Атрибут     | Нижче граничне значення | Оптимальне значення | Верхнє граничне значення | Дії                   |
|---------------------------|-------------|-------------------------|---------------------|--------------------------|-----------------------|
| pH Грунту для соняшника   | pH Грунту   | 6                       | 6,4                 | 6,8                      | Редагувати   Видалити |
| Ph. фосфору для соняшника | Ph. фосфору | 3                       | 7                   | 12                       | Редагувати   Видалити |
| Ph. калію для соняшника   | Ph. калію   | 9                       | 22                  | 34                       | Редагувати   Видалити |
| Ph. азоту для соняшника   | Ph. азоту   | 10                      | 25                  | 50                       | Редагувати   Видалити |

Рис. 23 Перелік радників на основі цифрових типів

Третім блоком сторінки є блок радників на основі перелічуваних типів. При створенні радника перелічуваного типу необхідно вказати назву та культуру, а лише потім задати відношення. Цей блок є більш складним – кожен рядок має стовпець відношень, які мають характер багато до одного. Відношення не можна редагувати, а лише видалити. Усього в таблиці чотири стовпці:

- Назва радника
- Атрибут радника
- Відношення
- Стовпець дій

При редагуванні радника перелічуваного типу не можна змінювати атрибут – бо це зробить відношення неактуальними, що не має сенсу з функціональної точки зору. При видаленні радника перелічуваного типу автоматично видаляється усі відношення для нього. У разі якщо усі відношення були сконфігуровані, рядок створення нового відношення не відображається. У разі якщо лише частина відношень є сконфігурованими, вже додані відношення є скритими з поля вибору значення перелічуваного типу.

Приклад конфігурації перелічуваного радника зображено на рис. 25.

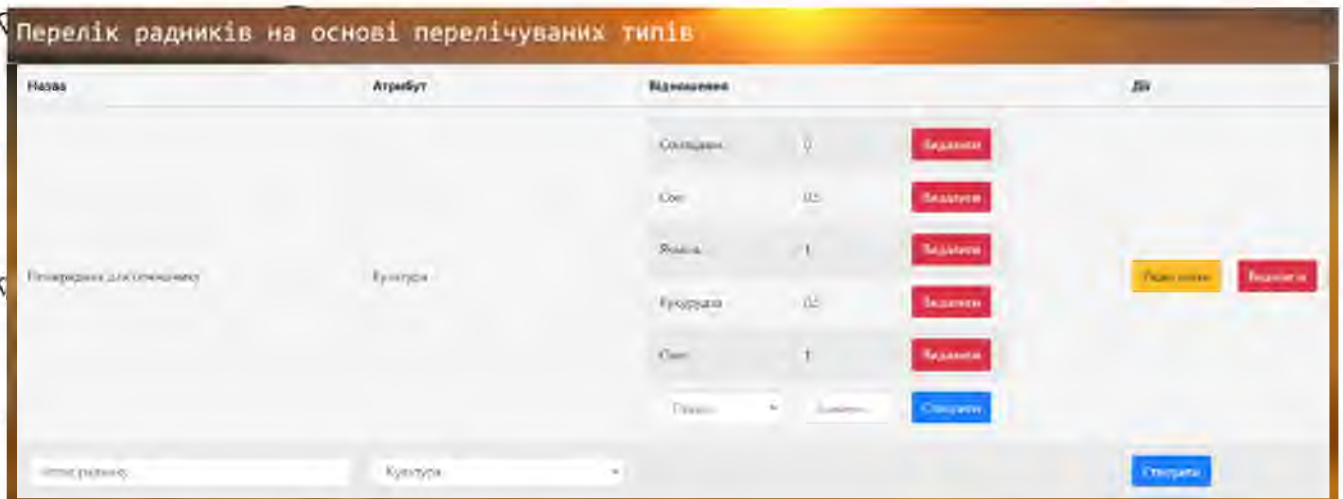


Рис. 24 Перелік радників на основі перелічуваних типів

#### 4.4 Розгортання системи

Для розгортання додатку пропонуються наступні умови:

- Сервер розміщується на машині на базі ОС Linux.
- База даних розміщується на хмарному середовищі AWS.
- Клієнтська частина додатку виконується у браузері користувача.
- Комунікація між сервером та базою даних відбувається через TCP/IP з застосуванням протоколу специфічного для PostgreSQL.
- Клієнт та сервер комунікують за протоколом HTTP та використовують формат даних JSON.

Linux – загальна назва UNIX-подібних операційних систем на основі однойменного ядра. Це один із найвидатніших прикладів розробки вільного та відкритого (з відкритим кодом, open source) програмного забезпечення. На відміну від власницьких операційних систем (на кшталт Microsoft Windows та MacOS X), їхні вихідні коди доступні всім для використання, зміни та поширення абсолютно вільно (в тому числі безкоштовно) [22].

Для розгортання доречно використовувати хмарну базу даних. Клієнтська частина додатку має виконуватись у веб-браузері користувача додатку.

Amazon Web Services (AWS) це безпечна платформа хмарних послуг, що пропонує обчислювальні потужності, сховища баз даних, доставку контенту та інші функціонал, який допомагає бізнесу масштабуватись і зростати [23].

До сервера поставлені наступні вимоги:

- Сервер має підтримувати віртуальну машину Java.
- У випадках високого навантаження на сервер, буде доречним встановити проміжний сервер який буде балансувати трафік між декількома копіями додатку.
- Сервер має мати не менше як 384 мегабайти оперативної пам'яті.
- Сервер повинен мати доступ до інтернету.
- На сервері має бути встановленим пакет OpenJDK 11.

Для розгортання додатку описаним вище способом, необхідно виконати наступні дії з боку адміністратора системи:

- Орендувати базу даних у хмарного сервісу та отримати ключі доступу до неї для подальших налаштувань.
- Створити базу даних 'fieldanalyzer' у PostgreSQL з кодуванням юнікод.
- Створити користувача бази даних та надати йому повні права на базу даних 'fieldanalyzer'.
- Орендувати віртуальний сервер у хмарного сервісу чи використати свій.
- Завантажити на сервер компонент OpenJDK 11.

НУБІП УКРАЇНИ

- Завантажити на віртуальний сервер .jar файл серверної частини додатку.
- Налаштувати автозапуск через додання серверної частини як сервісу у ОС Linux, та надати параметри доступу до бази даних.

НУБІП УКРАЇНИ

- Перезавантажити сервер Linux.

З метою підвищення безпеки та зручності використання додатку можна виконати наступні дії:

НУБІП УКРАЇНИ

- Додати доменне ім'я до серверу для зручності доступу з боку користувачів.
- Встановити TLS сертифікат для захисту від перехвату трафіку з боку кіберзлочинців.

НУБІП УКРАЇНИ

- Замовити послуги CDN-мережі для підвищення швидкості відклику серверу на запити від клієнта.
- Встановити SSH-ключ для доступу до сервера Linux для захисту від атак типу перебору пароля.

НУБІП УКРАЇНИ

Запропонований варіант розгортання додатку на основі операційної системи Linux візуалізовано на діаграмі розгортання на рис. 25. Зауважимо, що система підтримує інші варіанти розгортання – найбільш ключовими умовами є підтримка JVM, підтримка PostgreSQL та достатня кількість оперативної пам'яті та ресурсів центрального процесора.

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ



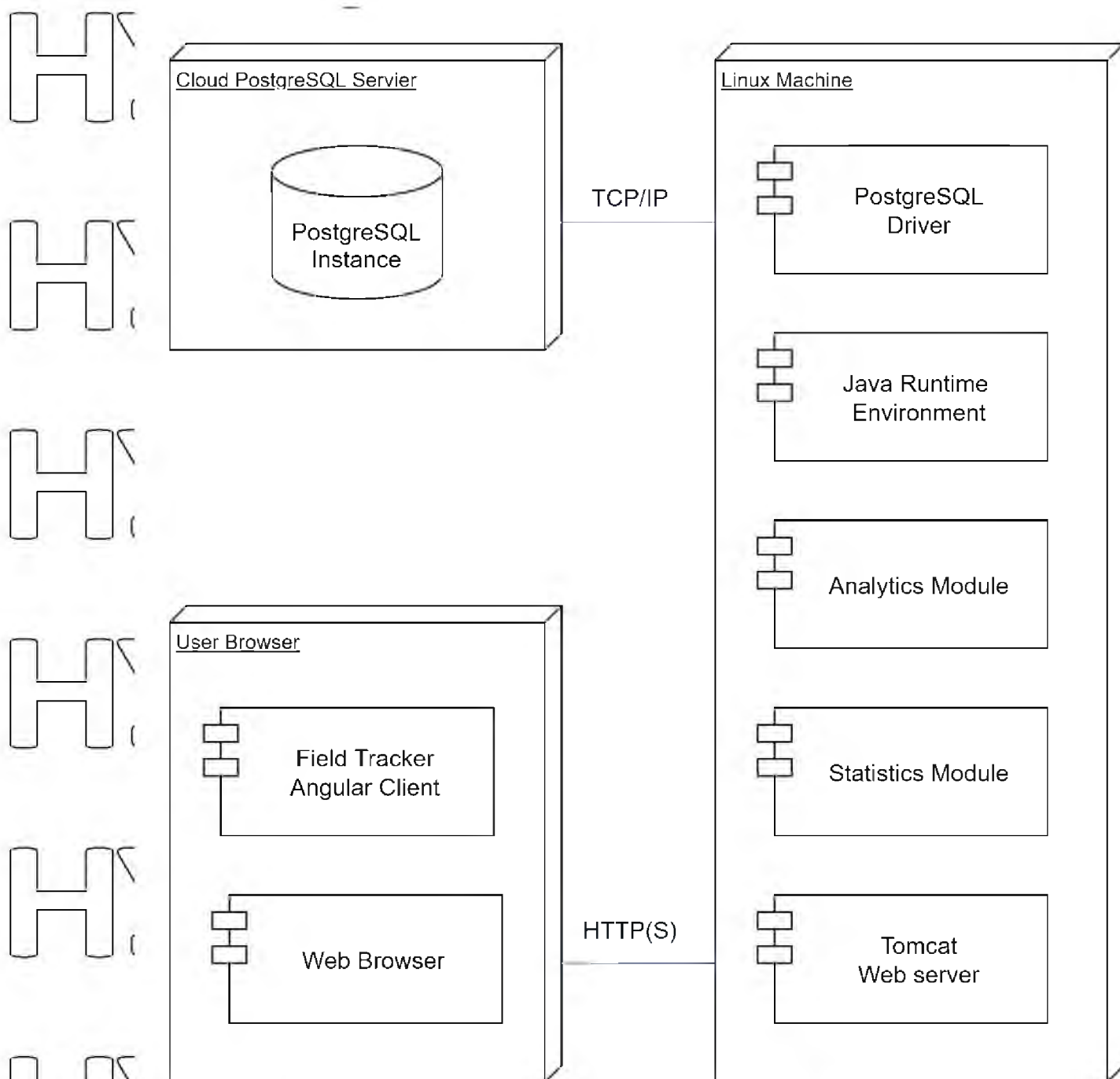


Рис. 25 Діаграма розгортання

НУБІП України

## 9. ВИСНОВКИ

За результатами виконання цієї роботи було проаналізовано предметну область планування сівозмін та розроблено аналітичний та статистичний модулі для автоматичного аналізу та підтримки прийняття рішень забезпечення врожайності сільськогосподарських культур агрофірми.

За результатами проведених досліджень, було зроблено висновок що:

- Використання сівозмін значно підвищує врожайність С/С культур.
- Для досягнення високої врожайності сільськогосподарських культур є важливими такі фактори, як: тип ґрунту, попередник, строк повернення на попереднє поле, кислотність ґрунту, вміст у ґрунті поживних речовин відповідної групи, тощо.
- Використання інформаційних технологій дозволить значно спростити процес планування сівозмін, підвищити врожайність, запобігти спустошенню земель, та скоротити використання мінеральних добрив.

Відповідно до проведених досліджень та сформульованої мети, були виконані наступні розробки:

- Було спроектовано систему, поставлені функціональні та нефункціональні вимоги, проаналізовані існуючі рішення, описані варіанти використання системи за допомогою діаграм UML, розроблено логічну модель бази даних модулів аналізу та статистики у вигляді ER-діаграми.
- Було реалізовано систему підтримки прийняття рішень на основі розробленої моделі та обраного інструментарію, описано процес розробки окремих частин систем аналізу та статистики та зображено створений інтерфейс користувача системи.

НУБІП України

- Запропоновано варіант розгортання системи у кінцевого користувача на базі ОС Linux та використанням бази даних у хмарному середовищі.

Отримані результати можуть бути використані для подальшого розвитку

класу систем підтримки прийняття рішень, що фокусуються на підвищенні якості

НУБІП України

планування сівозмін, та як основа для подальшого розвитку систем такого типу за рахунок урахування більшої кількості параметрів полів, прогнозування

температурних умов чи використання штучного інтелекту для покращення процесу

планування сівозмін.

НУБІП України

Розроблена система дозволяє просунути рівень автоматизації процесу прийняття рішень в області планування сівозмін, підвищити врожайність

сільськогосподарських культур агрофірми, та покращити можливості по боротьбі зі

спустошенням земель та надмірним використанням мінеральних добрив.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

## 10. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стецишин П. О. Пиндус В. В. Основи органічного виробництва.. Навчальний посібник, 2008. 117 с.
2. Система динамічної сівозміни. Агробізнес Сьогодні: веб-сайт. URL: <http://agro-business.com.ua/agro/ahronomiia-sohodni/item/10145-systema-dynamičnoi-s-vozminy.html> (дата звернення: 02.05.2018)
3. Структура посівів. Агробізнес Сьогодні: веб-сайт. URL: <http://agro-business.com.ua/agro/ahronomiia-sohodni/item/137-struktura-posiviv.html> (дата звернення: 04.05.2011)
4. Що таке OneSoil? Поради та відповіді від команди OneSoil URL: <https://intercom.help/onesoil/ru/articles/5237693-что-такое-onesoil>
5. Тестування платформи для моніторингу полей EOS Crop Monitoring. AGGEEK. URL: <https://aggeek.net/ru-blog/testirovanie-platforny-dlya-monitoringa-polej-eos-crop-monitoring> (дата звернення: 01.06.2020)
6. Логічна модель даних. Wikipedia. URL: [https://uk.wikipedia.org/wiki/Логічна\\_модель\\_даних](https://uk.wikipedia.org/wiki/Логічна_модель_даних)
7. Реляційна база даних. UA5.org. URL: <https://ua5.org/database/189-reliaciina-baza-danikl.html>
8. Бази даних: конспект лекцій. Google Books. URL: <https://shorturl.at/ekERU>
9. Введення. SwimLane Gitbook. URL: <https://swimlane.gitbook.io/ngx-charts/>
10. Повний посібник із діаграм з областями. Chartio. URL: <https://chartio.com/learn/charts/area-chart-complete-guide/>
11. JSON. Wikipedia. URL: <https://uk.wikipedia.org/wiki/JSON>
12. SVG Документація. Mozilla MDN Web Docs. URL: <https://developer.mozilla.org/ru/docs/Web/SVG>
13. Мова програмування Java та платформа JavaFX. Приклади застосування. Державний Університет Телекомунікацій. URL:

<http://www.dit.edu.ua/ua/news-1-626-6031-mova-programuvannya-java-ta-platforma-javafx-prikladi-zastosuvannya-kafedra-kompyuternih-nauk-ta-informacynih-tehnologiy>

14. Craig Walls. Spring in Action, Fifth Edition. с. 4

15. Hibernate. Wikipedia. URL: <https://uk.wikipedia.org/wiki/Hibernate>

16. Apache Tomcat. Apache. URL: <http://tomcat.apache.org>

17. Введення в Maven – простий інструмент автоматизації збірки. itGap, URL:

<https://itgap.ru/post/vvedenie-v-maven>

18. JetBrains IntelliJ IDEA. Softico. URL: <https://softico.ua/uk/jetbrains-intellij-idea/>

19. Bootstrap. Зіна дизайн. URL: <https://zina.design/uk/dictionary/bootstrap/>

20. JavaScript that scales. TypeScript. URL: <https://www.typescriptlang.org>

21. Введення в Angular, Що таке Angular. Перший проект. Metanit. URL:

<https://metanit.com/web/angular2/1-1.php>

22. Linux. Wikipedia. URL: <https://uk.wikipedia.org/wiki/Linux>

23. Що таке Amazon Web Services?. Techsoup Україна. URL:

<https://ukraine.techsoup.global/taxonomy/term/279/faq>

НУБІП України

НУБІП України

НУБІП України

НУБІП України **11. ДОДАТОК А**

НУБІП України

НУБІП України

**DDL скрипт ініціалізації бази даних**

НУБІП України

НУБІП України

Сторінок 2

НУБІП України

НУБІП України

```
CREATE TABLE IF NOT EXISTS account
```

```
(
  account_id          SERIAL PRIMARY KEY NOT NULL,
  name                VARCHAR(32) UNIQUE NOT NULL,
  password_hash       TEXT NOT NULL,
  anchor_latitude     TEXT DEFAULT '50.4019514' NOT NULL,
  anchor_longitude    TEXT DEFAULT '30.3926095' NOT NULL,
  disabled            BOOLEAN DEFAULT FALSE NOT NULL,
  role                TEXT NOT NULL CHECK
  ( role IN ('ROLE_ADMIN', 'ROLE_AGRONOMIST', 'ROLE_MANAGER', 'ROLE_SPECIALIST') )
);;
```

```
CREATE TABLE IF NOT EXISTS area
```

```
(
  area_id             SERIAL PRIMARY KEY NOT NULL,
  point1              PATH NOT NULL,
  point2              PATH NOT NULL,
  point3              PATH NOT NULL,
  point4              PATH NOT NULL
);;
```

```
CREATE TABLE IF NOT EXISTS field_group
```

```
(
  group_id            SERIAL PRIMARY KEY NOT NULL,
  name                VARCHAR(32) NOT NULL,
  description          TEXT,
  account_id          INTEGER REFERENCES account (account_id) ON DELETE CASCADE
);;
```

```
CREATE TABLE IF NOT EXISTS field
```

```
(
  field_id             SERIAL PRIMARY KEY NOT NULL,
  name                 VARCHAR(32) NOT NULL,
  description          TEXT,
  color                TEXT CHECK ( color LIKE '#_____' ),
  border_color         TEXT CHECK ( color LIKE '#_____' ),
  group_id             INTEGER REFERENCES field_group (group_id) ON DELETE SET DEFAULT,
  area_id              INTEGER REFERENCES area (area_id) ON DELETE CASCADE,
  account_id           INTEGER REFERENCES account (account_id) ON DELETE CASCADE,
  disabled             BOOLEAN DEFAULT FALSE NOT NULL
);;
```

```
CREATE TABLE IF NOT EXISTS field_attribute
```

```
(
  attribute_id         SERIAL PRIMARY KEY NOT NULL,
  name                 VARCHAR(32) NOT NULL,
  description          TEXT,
  type                 TEXT CHECK ( type = 'NUMERIC' OR type = 'TEXTUAL' OR type = 'ENUM' ) NOT NULL,
  static              BOOLEAN NOT NULL,
  readonly             BOOLEAN NOT NULL DEFAULT FALSE,
  system_key          TEXT UNIQUE
);;
```

```
CREATE TABLE IF NOT EXISTS field_enum
```

```
(
  enum_id SERIAL PRIMARY KEY,
  value VARCHAR(32) NOT NULL,
  attribute_id INTEGER REFERENCES field_attribute (attribute_id) ON DELETE RESTRICT
);;
```

```
CREATE TABLE IF NOT EXISTS field_advisor_numeric
```

```
(
  advisor_id SERIAL PRIMARY KEY,
  attribute_id INTEGER REFERENCES field_attribute (attribute_id) ON DELETE RESTRICT,
  advisee_id INTEGER REFERENCES field_enum(enum_id) ON DELETE RESTRICT,
  name TEXT NOT NULL,
  lower_non_optimal_value NUMERIC(12, 2),
  optimal_value NUMERIC(12, 2),
  upper_non_optimal_value NUMERIC(12, 2)
);;
```

```
CREATE TABLE IF NOT EXISTS field_advisor_enum
```

```
(
  advisor_id SERIAL PRIMARY KEY,
  attribute_id INTEGER REFERENCES field_attribute (attribute_id) ON DELETE RESTRICT,
  advisee_id INTEGER REFERENCES field_enum(enum_id) ON DELETE RESTRICT,
  name TEXT NOT NULL
);;
```

```
CREATE TABLE IF NOT EXISTS field_advisor_enum_relation
```

```
(
  relation_id SERIAL PRIMARY KEY,
  advisor_id INTEGER REFERENCES field_advisor_enum (advisor_id) ON DELETE RESTRICT,
  enum_ref INTEGER REFERENCES field_enum(enum_id) ON DELETE CASCADE,
  relation_value NUMERIC(12, 2) CHECK (relation_value BETWEEN 0 AND 1) NOT NULL
);;
```

```
CREATE TABLE IF NOT EXISTS field_history
```

```
(
  history_id SERIAL PRIMARY KEY NOT NULL,
  field_id INTEGER REFERENCES field (field_id) ON DELETE CASCADE,
  date_to DATE NOT NULL,
  note TEXT,
  date_from DATE NOT NULL
);;
```

```
CREATE TABLE IF NOT EXISTS field_parameter
```

```
(
  param_id SERIAL PRIMARY KEY NOT NULL,
  value TEXT,
  enum_ref INTEGER REFERENCES field_enum (enum_id) ON DELETE CASCADE,
  attribute_id INTEGER REFERENCES field_attribute (attribute_id) ON DELETE CASCADE,
  history_id INTEGER REFERENCES field_history (history_id) ON DELETE CASCADE,
  UNIQUE (attribute_id, history_id)
);;
```

```
CREATE TABLE IF NOT EXISTS field_property
```

```
(
  property_id SERIAL PRIMARY KEY NOT NULL,
  value TEXT,
  enum_ref INTEGER REFERENCES field_enum (enum_id) ON DELETE CASCADE,
  attribute_id INTEGER REFERENCES field_attribute (attribute_id) ON DELETE CASCADE,
  field_id INTEGER REFERENCES field (field_id) ON DELETE CASCADE,
  UNIQUE (attribute_id, field_id)
);;
```