**Програмна система для моніторингу стану рослин**

1 Вступ…………………………………………………...………………….2

1.1 Огляд програмного забезпечення.………………………………2

1.2 Мета роботи.……………………………………………………...2

1.3 Межі……….………………………………………………………2

2 Загальний опис…………………………………………………………….3

2.1 Перспективи продукту………………………...…………………3

2.2 Функції продукту……………………………...………………….3

2.3 Характеристики користувачів……………………………………3

2.4 Загальні обмеження…………………………...………………….3

2.5 Припущення та залежності………………………………....……4  
3 Конкретні вимоги………………………………………………………….5

3.1 Вимоги до зовнішніх інтерфейсів……………………………….5

3.1.1 Інтерфейс користувача…………………………………..5  
 3.1.2 Програмний інтерфейс…………………………………..5

3.1.3 Комунікаційний інтерфейс...…………………………….5

3.1.4 Обмеження пам’яті………………………………………5  
 3.1.4 Обмеження пам’яті………………………………………5

3.1.5 Функції продукту.………………………………………..5  
 3.1.6 Припущення та залежності……………………………..5

3.2 Властивості програмного продукту……………………………..6  
 3.3 Атрибути програмного продукту………………………………..6

3.3.1 Надійність………………………………………………..6  
 3.3.2 Доступність………………………………………………6

3.3.3 Безпека……………………………………………………6  
 3.3.4 Супроводжуваність………………………………………6  
 3.3.5 Переносимість……………………………………………6

3.3.6 Продуктивність…………………………………………..6  
3.4 Вимоги бази даних……………………………………………….7

**1 ВСТУП**

1.1 Огляд програмного забезпечення  
 Розроблене програмне забезпечення являє собою автоматизовану інформаційну систему для моніторингу мікроклімату теплиць у реальному часі. Система реалізована у вигляді клієнт-серверного застосунку з використанням сучасного стеку технологій: Java (Spring Boot) для серверної частини, PostgreSQL для зберігання даних, MQTT для обміну повідомленнями між сенсорами та сервером, React для реалізації користувацького інтерфейсу.

1.2 Мета роботи  
 Метою є створення надійної, масштабованої та зручної у використанні системи для моніторингу стану теплиць, що дозволяє збирати, зберігати, аналізувати та візуалізувати сенсорні дані, а також надсилати сповіщення та рекомендації користувачам.

1.3 Межі  
 Система не здійснює автоматичне керування теплицею (наприклад, увімкнення поливу чи вентиляції). Вона орієнтована на моніторинг, сповіщення та надання рекомендацій на основі зібраних даних. Робота з мобільними пристроями реалізована лише у вигляді адаптивної верстки.

**2 ЗАГАЛЬНИЙ ОПИС**

2.1 Перспективи продукту  
 Система може бути інтегрована з іншими агротехнологічними рішеннями або масштабована до SaaS-платформи для обслуговування багатьох користувачів. Подальший розвиток передбачає додавання елементів керування обладнанням, мобільного застосунку та розширення аналітичного модуля.

2.2 Функції продукту

* реєстрація та авторизація користувачів з розмежуванням ролей;
* Додавання теплиць, сенсорів та рослин;
* отримання даних від сенсорів у реальному часі через MQTT;
* Збереження та обробка історичних даних у PostgreSQL;
* відображення актуальних показників та графіків у веб-інтерфейсі;
* формування системи рекомендацій на основі граничних параметрів культур;
* надсилання сповіщень про перевищення норм або критичні стани.

2.3 Характеристики користувачів  
 Системою можуть користуватись агрономи, власники теплиць, технічний персонал. Мінімальні вимоги до користувача — базові навички роботи з веб-браузером.

2.4 Загальні обмеження

* MQTT-брокер має бути доступним для сенсорів та сервера;
* сенсорне обладнання має підтримувати протокол MQTT;
* доступ до веб-інтерфейсу можливий лише за умови наявності інтернет-з’єднання.

2.5 Припущення та залежності

* усі сенсори передають дані у відповідному форматі (JSON);
* сервер має постійне з’єднання з базою даних;
* клієнтська частина отримує дані через REST API та WebSocket (для MQTT-обновлень).

**3 КОНКРЕТНІ ВИМОГИ**

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача  
 Інтерфейс реалізований за допомогою React. Передбачено форми для реєстрації, входу, додавання теплиць, відображення показників у вигляді графіків, таблиць та повідомлень. Інтерфейс адаптивний, працює на ПК та мобільних пристроях.

3.1.2 Програмний інтерфейс  
 REST API реалізовано за допомогою Spring Boot. Передбачені маршрути для CRUD-операцій над користувачами, теплицями, рослинами, сенсорами, історією показників, а також маршрут для автентифікації з JWT.

3.1.3 Комунікаційний інтерфейс  
 MQTT використовується для передачі даних із сенсорів на сервер. Сервер підписаний на відповідні топіки та обробляє повідомлення у форматі JSON.

3.1.4 Обмеження пам’яті  
 Не визначено жорстких обмежень, однак рекомендується розгортати систему на сервері з 2 ГБ ОЗП та не менше 2 ГБ вільного місця під базу даних.

3.1.5 Функції продукту  
 Перелік функцій відповідає розділу 2.2: CRUD-операції, обробка MQTT, візуалізація, сповіщення, авторизація.

3.1.6 Припущення та залежності

* користувачі мають унікальні email-адреси;
* всі дії над об’єктами можливі лише після автентифікації.

3.2 Властивості програмного продукту  
 Система модульна, реалізована з розділенням на клієнтську, серверну частину та брокер повідомлень. Серверна частина ізольована за допомогою REST API, що дозволяє легко інтегрувати додаткові інтерфейси (наприклад, мобільні додатки).

3.3 Атрибути програмного продукту

3.3.1 Надійність  
 Система протестована в умовах симуляції даних із сенсорів. Передбачено логування, обробку помилок з повідомленням користувача.

3.3.2 Доступність  
 Система доступна через веб-браузер 24/7, за умови роботи сервера та наявності інтернет-з’єднання.

3.3.3 Безпека  
 Дані користувачів захищені через JWT-автентифікацію та шифрування паролів (bcrypt). Доступ до ресурсів обмежений за ролями.

3.3.4 Супроводжуваність  
 Код структурований, розділений на окремі модулі. Документовані API, структура бази даних і конфігурації дозволяють легко масштабувати та змінювати систему.

3.3.5 Переносимість  
 Система може бути розгорнута у будь-якому середовищі, що підтримує Java та PostgreSQL (локально або у хмарі).

3.3.6 Продуктивність  
 Завдяки використанню MQTT система здатна обробляти велику кількість сенсорних повідомлень у реальному часі без суттєвого навантаження на сервер.

3.4 Вимоги бази даних  
 База даних реалізована у PostgreSQL. Схема нормалізована до третьої нормальної форми. Використовуються зовнішні ключі для забезпечення цілісності. Дані зберігаються у п’яти основних таблицях: користувачі, теплиці, рослини, сенсори, історія станів сенсорів, а також таблиця сповіщень.