Специфікація ПЗ   
  
Міністерство освіти і науки України   
Харківський національний університет радіоелектроніки   
  
Факультет комп’ютерних наук   
Кафедра програмної інженерії   
  
  
  
  
СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ   
веб застосунок для обліку та планування замовлень по догляду за садом та газоном “GreenFlow”.  
  
  
Студент гр. ПЗПІ-21-7 \_\_\_\_\_\_\_\_\_\_\_ Кошель Д.Ю.   
Студент гр. ПЗПІ-21-7 \_\_\_\_\_\_\_\_\_\_\_ Пєхотін А.Ю.

Харків 2024 р.

ЗМІСТ

[1 Вступ 3](#_Toc200401915)

[1.1 Огляд продукту 3](#_Toc200401916)

[1.2 Мета 3](#_Toc200401917)

[1.3 Межі 3](#_Toc200401918)

[1.4 Посилання 4](#_Toc200401919)

[1.5 Означення та абревіатури 4](#_Toc200401920)

[2 Загальний опис 5](#_Toc200401921)

[2.1 Перспективи продукту 5](#_Toc200401922)

[2.2 Функції продукту 5](#_Toc200401923)

[2.3 Характеристики користувачів 6](#_Toc200401924)

[2.4 Загальні обмеження 6](#_Toc200401925)

[2.5 Припущення й залежності 6](#_Toc200401926)

[3 Конкретні вимоги 8](#_Toc200401927)

[3.1 Вимоги до зовнішніх інтерфейсів 8](#_Toc200401928)

[3.2 Властивості програмного продукту 10](#_Toc200401929)

[3.4 Вимоги бази даних 11](#_Toc200401930)

[3.5 Інші вимоги 11](#_Toc200401931)

1 Вступ

### Огляд продукту

Програмне забезпечення, що описується у цій специфікації, є комплексною інформаційною системою для керування процесами у сфері догляду за садами та газонами (GreenFlow). Система включає в себе веб-застосунок для замовників, що дозволяє створювати та контролювати виконання замовлень, а також внутрішній інтерфейс для працівників, які можуть обирати, виконувати та закривати завдання. Основний акцент зроблено на автоматизацію процесів, географічну фільтрацію замовлень, оновлення фінансових балансів і підтримку онлайн-оплати.

### 1.2 Мета

Метою створення даної системи є підвищення ефективності організації послуг із догляду за зеленими територіями шляхом автоматизації процесів замовлення, призначення виконавців, виконання робіт та ведення фінансового обліку. Програмне забезпечення повинно забезпечити зручний та надійний механізм взаємодії між замовниками та виконавцями, мінімізувати людський фактор і підтримувати масштабованість.

### 1.3 Межі

Розроблювана система охоплює лише цифрову частину організації процесу — веб-інтерфейси для клієнтів та працівників, API для взаємодії між мікросервісами, бекенд-логіку, роботу з геоданими, балансами, повідомленнями та оплатами. До сфери відповідальності не входить безпосереднє фізичне виконання робіт, закупівля інвентарю або кадрове управління. У межах роботи не передбачається мобільна версія, однак система повинна бути адаптованою для подальшого розширення.

### 1.4 Посилання

1. IEEE Std 830-1998 – Recommended Practice for Software Requirements Specifications

2. Spring Boot Documentation:<https://docs.spring.io/spring-boot/docs/current/reference/html/>

3. React Documentation:<https://react.dev/>

4. OpenAPI Specification: https://swagger.io/specification/

### 1.5 Означення та абревіатури

GreenFlow - назва системи, що розробляється

API (Application Programming Interface) – інтерфейс програмного застосування

UI (User Interface) – інтерфейс користувача

CRUD (Create, Read, Update, Delete) – базові операції з даними

JWT (JSON Web Token) – стандарт передачі авторизаційної інформації

OAuth2 - Протокол авторизації, що дозволяє обмежений доступ до ресурсів

PayPal - сервіс онлайн-платежів, інтегрований для обробки транзакцій

2 Загальний опис

### 2.1 Перспективи продукту

GreenFlow є самостійною багаторівневою веб-системою, яка функціонує як платформа для замовлення, управління та виконання послуг з догляду за зеленими насадженнями. Система включає окремі ролі для замовників та виконавців. Програмне забезпечення розробляється з урахуванням майбутнього масштабування, з можливістю інтеграції з мобільним додатком, розширенням карти покриття, підключенням додаткових платіжних сервісів та запровадженням автоматизованих розкладів робіт.

### 2.2 Функції продукту

Має такі основні функції:

* створення та перегляд замовлень користувачем;
* реєстрація, автентифікація та авторизація;
* створення та редагування інформації про сад;
* завантаження до 12 зображень до кожного саду;
* автоматична перевірка дати відкриття та закриття замовлення;
* пошук замовлень виконавцями на основі геолокації;
* створення та редагування обладнання;
* перегляд списків обладнання, садів, замовлень.
* перегляд інформації про себе, редагування контактних даних;
* вибір і закриття замовлень працівниками;
* автоматичне оновлення фінансових балансів користувачів та працівників;
* фільтрація списку обладнання за ціною, назвою тощо;
* поповнення балансу через інтеграцію з платіжною системою PayPal;
* надсилання email-сповіщень;

### 2.3 Характеристики користувачів

Система розрахована на такі категорії користувачів:

* змовники (клієнти) – фізичні або юридичні особи, які замовляють послуги через веб-інтерфейс. Мають базовий рівень цифрової грамотності;
* виконавці (працівники) – зареєстровані співробітники, що отримують доступ до замовлень відповідно до географічної зони;
* менеджери – відповідальні за управління складами з обладнанням.

### 2.4 Загальні обмеження

* система доступна лише у веб-версії через сучасні браузери (Google Chrome, Firefox, Edge);
* виконання геопошуку та відображення замовлень обмежене межами 100 км від поточного місцезнаходження користувача;
* оплата доступна лише через інтеграцію з PayPal (інші методи не підтримуються на поточному етапі);
* всі грошові операції проводяться у валюті Euro;
* система не підтримує офлайн-режим;
* функціональність розрахована на середнє навантаження (до 500 активних користувачів одночасно).
* доступ до геолокації залежить від дозволу користувача у браузері;
* завантаження зображень відбувається через форму з перевіркою MIME-типу файлу.

### 2.5 Припущення й залежності

* передбачається наявність стабільного інтернет-з’єднання у всіх користувачів;
* користувач надає браузеру дозвіл на доступ до геолокації для коректної роботи функцій пошуку
* користувач самостійно вводить координати саду.
* усі працівники мають доступ до пристрою з GPS для коректного визначення місцезнаходження;
* сторонні сервіси (PayPal, OpenStreetMap або Google Maps) залишаються стабільними та доступними;
* майбутня мобільна версія буде побудована на тих самих API, що й веб-інтерфейс.

3 Конкретні вимоги

### 3.1 Вимоги до зовнішніх інтерфейсів

#### 3.1.1 Інтерфейс користувача

* графічний інтерфейс реалізований на базі React;
* інтерфейс має бути реалізований як SPA (Single Page Application) із динамічною навігацією без перезавантаження сторінки;
* інтерфейс має бути інтуїтивно зрозумілим, забезпечувати легкий доступ до всіх основних функцій;
* усі компоненти мають бути локалізовані українською мовою;
* всі обов’язкові поля повинні бути чітко позначені (\* або іншим способом);
* інтерфейс повинен бути адаптивним до екранів ноутбуків і великих моніторів;
* після виконання ключових дій (створення саду, підтвердження оренди, оновлення даних) користувач має отримувати повідомлення про успішне виконання або помилку;
* дизайн інтерфейсу має відповідати сучасним вимогам UI/UX: чітка сітка, мінімалістичні іконки, достатні відступи та зрозуміла типографіка;
* усі форми мають валідацію на стороні клієнта: перевірка обов’язкових полів, перевірка формату електронної пошти, числових значень тощо.

#### 3.1.2 Апаратний інтерфейс

Система не має специфічних апаратних залежностей. Підтримується використання GPS-модулів виконавців для визначення їх місцезнаходження.

#### 3.1.3 Програмний інтерфейс

* RESTful API для взаємодії між frontend і backend;
* запити на сервер за допомогою бібліотеки Axios;
* авторизація через власну форму логіну;
* стандартний інтерфейс геолокації Web API (navigator.geolocation);
* ендпоінти для роботи з замовленнями, користувачами, платежами, повідомленнями та фінансовими операціями;

#### 3.1.4 Комунікаційний протокол

* вся взаємодія відбувається через HTTPS із використанням JWT-токенів;
* API Gateway забезпечує маршрутизацію запитів до мікросервісів.

#### 3.1.5 Обмеження пам’яті

* обмеження пам’яті залежать від середовища деплою. У межах одного мікросервісу – до 512 МБ оперативної пам’яті за замовчуванням, чого достатньо для тестової роботи з мінімальним навантаженням.
* файли (фотографії) зберігаються в об’єктному сховищі DigitalOcean Spaces.

#### 3.1.6 Операції

* створення, редагування, перегляд та видалення замовлень;
* додавання та видалення обладнання, зміна статусу оренди;
* вибір замовлення працівником та оновлення його статусу;
* проведення оплати через PayPal;
* надсилання email листів.

#### 3.1.7 Функції продукту

* облік і перегляд фінансових балансів;
* обмеження доступу до замовлень за геозоною;
* генерація автоматичних звітів по завершеним роботам;
* верифікація працівників адміністратором.

#### 3.1.8 Припущення й залежності

* підключення до PayPal API;
* зовнішні сервіси геолокації (OpenStreetMap / Google Maps) працюють стабільно;
* сторонні бібліотеки front-end (наприклад, React Leaflet) використовуються без змін.

### 3.2 Властивості програмного продукту

* підтримка масштабування в Kubernetes середовищі;
* робота з розподіленою архітектурою мікросервісів;
* сумісність з сучасними браузерами;
* прозора обробка помилок з повідомленнями користувачеві.

3.3 Атрибути програмного продукту

#### 3.3.1 Доступність

* цілодобовий доступ через веб-браузер;
* мінімальний даунтайм завдяки Kubernetes з автоматичним перезапуском контейнерів.

#### 3.3.2 Безпека

* HTTPS-з’єднання, JWT-токени, обмеження доступу за ролями;
* валідація даних на бекенді;
* захист від SQL-ін’єкцій та XSS;
* при втраті сесії користувача має бути автоматично переадресовано на сторінку логіну.

#### 3.3.4 Супроводжуваність

* код структурований за принципами SOLID та Clean Architecture;
* всі мікросервіси мають юніт- і інтеграційні тести;

#### 3.3.5 Переносимість

* сервіси запускаються в Docker-контейнерах;
* використання стандартного API та конфігурацій через .env.

#### 3.3.6 Продуктивність

* асинхронна обробка повідомлень через RabbitMQ;
* нормалізована структура БД.

### 3.4 Вимоги бази даних

3.4.1 Реляційна база PostgreSQL.

* основні таблиці: users, gardens, orders, user-balance, payments, equipment;
* доступ до БД захищений паролем, доступ має лише бекенд;

3.4.1 Нереляційні сховища

* Redis для збереження відкритих замовлень для швидкого гео-пошуку;
* MongoDB для збереження неструктурованих даних про обладнання.

### 3.5 Інші вимоги

* підтримка багатомовності (на етапі MVP — лише англійська);
* можливість додавання фото до садів та газонів;
* підтримка CI/CD через GitHub Actions.