



Spis treści

Model opisowy	2
Project Charter.....	2
Cel projektu.....	2
Zakres (wstępny).....	3
Wskaźniki sukcesu (KPI/OKR).....	3
Założenia i ograniczenia	4
Interesariusze (lista wysokiego poziomu).....	4
Ryzyka wstępne	4
Plan wysokopoziomowy	5
Aktorzy.....	5
Przypadki użycia (Use cases)	7
Diagram Use Case:	9
3. Specyfikacja wymagań.....	9
3.1 Wymagania funkcjonalne	9
3.2 Wymagania pozafunkcjonalne.....	10
Wybór technologii	10
Diagram UML	11
Diagram ERD.....	12
Podsumowanie	13

Model opisowy

Aplikacja „Oasis” w założeniu ma być aplikacją do monitorowania oraz regulowania parametrów fizycznych roślin. Ma to polegać na bieżącym wyświetlanie parametrów roślin i ich otoczenia takich jak: wilgotność i temperatura powietrza, wilgotność i pH gleby, ciśnienie atmosferyczne, nasłonecznienie. Dodatkowo, aplikacja ma posiadać możliwość kontroli nad nawodnieniem roślin poprzez jednorazowe podlanie w czasie podanym przez użytkownika lub ustawienie harmonogramu podlewania w trakcie dnia/tygodnia. Aplikacja powinna zezwalać użytkownikowi także na podgląd uprawy na żywo. Wszystkie te funkcjonalności wymagają, aby użytkownik dysponował odpowiednimi czujnikami do pomiaru parametrów oraz kamerami do podglądu na żywo. Aplikacja powinna pozwalać na edytowanie i usuwanie istniejących czujników/kamer wirtualnych (ich aplikacyjnej reprezentacji) oraz dodawanie nowych. Podobnie w przypadku harmonogramów podlewania: użytkownik powinien mieć możliwość edycji lub usunięcia istniejących harmonogramów jak dodanie nowych. W przypadku jednorazowego podlewania użytkownik powinien móc natychmiast przerwać proces podlewania. W ramach przypomnienia użytkownikowi o potrzebie podlania lub sprawdzenia obecnych parametrów, użytkownik może ustawić możliwość wysyłania powiadomień przez aplikację za pomocą poczty e-mail. Aby zapewnić bezpieczeństwo i rozdzielność pomiędzy użytkownikami aplikacja ma umożliwiać proces logowania i rejestracji do systemu wykorzystując aktualne algorytmy szyfrujące. Aplikacja powinna być responsywna i działać zarówno na urządzeniach stacjonarnych jak i mobilnych.

Project Charter

Cel projektu

Opis celu biznesowego, problemu i wartości dla użytkownika/organizacji.

Celem projektu jest stworzenie webowej aplikacji do monitorowania i automatycznego podlewania roślin, łączącej odczyty z czujników środowiskowych z kontrolą mikrokontrolera. Rozwiązanie ma zredukować ręczne czynności związane z podlewaniem i pielęgnacją uprawy dzięki zdalnemu monitorowaniu w czasie rzeczywistym. Użytkownicy - hodowcy amatorzy i małe gospodarstwa - otrzymają przejrzysty pulpit do śledzenia wilgotności, temperatury i natężenia światła oraz możliwość definiowania powiadomień i automatycznych akcji.

Zakres (wstępny)

Co wchodzi w zakres (MVP).

- Zarządzanie czujnikami i akcjami podlewania (operacje CRUD).
- Wyświetlanie paneli z wartościami parametrów
- Wyświetlanie przycisków ręcznego uruchamiania akcji
- Podstawowe logowanie/rejestrowanie
- Autoryzacja użytkowników (podział na właściciela uprawy, który ma uprawnienia do zarządzania czujnikami i kamerami oraz na pracownika/gościa – jedynie monitorowanie i podlewanie/planowanie).
- Wyświetlenie podglądu z kamer – transmisje live (kamery/transmisje dodane wcześniej)

Co jest poza zakresem (Out of scope).

- Rozbudowany system raportów historycznych i analizy trendów.
- Integracja z zewnętrznymi platformami IoT (np. Home Assistant).
- Skalowanie rozproszonej architektury
- Automatyczna diagnostyka sprzętu
- Przechowywanie i analiza nagrani z kamer
- Sterowanie ogrzewaniem, oświetleniem, wentylacją, nawożeniem

Wskaźniki sukcesu (KPI/OKR)

Wymień mierzalne cele i progi sukcesu.

- Liczba pomiarów poprawnie odebranych przez system $\geq 95\%$ dla 1000 prób.
- Liczba uschniętych/więdzących roślin w uprawie poniżej 5%.
- Czas od wysłania komendy podlewania natychmiastowego do wykonania tej akcji poniżej 5s w 90 % przypadków.
- Czas wstępnej instalacji i konfiguracji aplikacji ≤ 20 min dla nowego użytkownika.
- 100% operacji kończy się sukcesem w testach

Założenia i ograniczenia

Kluczowe założenia, zależności, ograniczenia czasowe/budżetowe.

- Zakładamy, że mikrokontroler jest wyposażony w działający klient MQTT lub moduł LoRa (na potrzeby MVP aplikacja będzie zbierała dane z gotowych API online).
- Budżet: jedynie koszty infrastruktury studenckiej (darmowe kontenery)
- Ograniczenie związane z brakiem/niekompatybilnością kamery
- Czas realizacji: październik 2025 – styczeń 2026.
- Zespół projektowy: 2 osoby, brak dedykowanego administratora sieci.
- Ograniczona wiedza końcowego użytkownika, interfejs musi być intuicyjny.
- Odpowiednio wyposażony użytkownik, który chce dołączyć do projektu: odpowiednie czujniki, mikrokontroler kompatybilny z docelowym protokołem przesyłania danych

Interesariusze (lista wysokiego poziomu)

Sponsor, kluczowi użytkownicy, zespoły, partnerzy.

- Prowadzący przedmiot (opiekun akademicki) - ocena techniczna i merytoryczna.
- Studenci (zespoł projektowy) - rozwój, testy, dokumentacja.
- Użytkownicy końcowi (hobbystyczni ogrodnicy) - akceptacja użyteczności i funkcji.
- Dostawca mikrokontrolera/czujników/kamer - wsparcie techniczne przy integracji.
- Wydział Informatyki i Matematyki (uczelnia) - prawo własności

Ryzyka wstępne

1. Problemy z połączeniem MQTT/LoRa.
2. Problemy w konfiguracji i kompatybilności czujników i mikrokontrolera, kamer.
3. Brak zasobów czasowych zespołu (kolizje z innymi zajęciami lub egzaminami).
4. Zmiany wymagań funkcjonalnych w trakcie semestru (zakres rozrośnie się poza MVP).
5. Zbyt złożone problemy, początkowo uznawane za trywialne

Plan wysokopoziomowy

Kamienie milowe i orientacyjne daty.

- Październik 2025: zebranie wymagań, research integracji MQTT/LoRa, przygotowanie specyfikacji architektury.
- Listopad 2025: implementacja frontendu MVP – widoki i stylowanie
- Grudzień 2025: tworzenie backendu i połączenia z bazą danych
- Styczeń 2026: pełne MVP z kilkoma feature'ami, produkt gotowy do oddania w ramach zaliczenia, pełna dokumentacja

Aktorzy

Nazwa	Opis	Rodzaj
Niezarejestrowany użytkownik	Może się zarejestrować w systemie.	Osoba
Zarejestrowany użytkownik (Właściciel uprawy)	Pełne uprawnienia: zarządzanie czujnikami, kamerami, harmonogramami, ręczne podlewanie, konfiguracja powiadomień.	Osoba
Zarejestrowany użytkownik (Pracownik/Gość)	Ograniczone uprawnienia: monitorowanie parametrów, podgląd kamer, uruchamianie podlewania, planowanie harmonogramów.	Osoba
Administrator systemu	Odpowiada za utrzymanie i konfigurację systemu, zarządzanie użytkownikami, monitorowanie poprawności działania.	Osoba

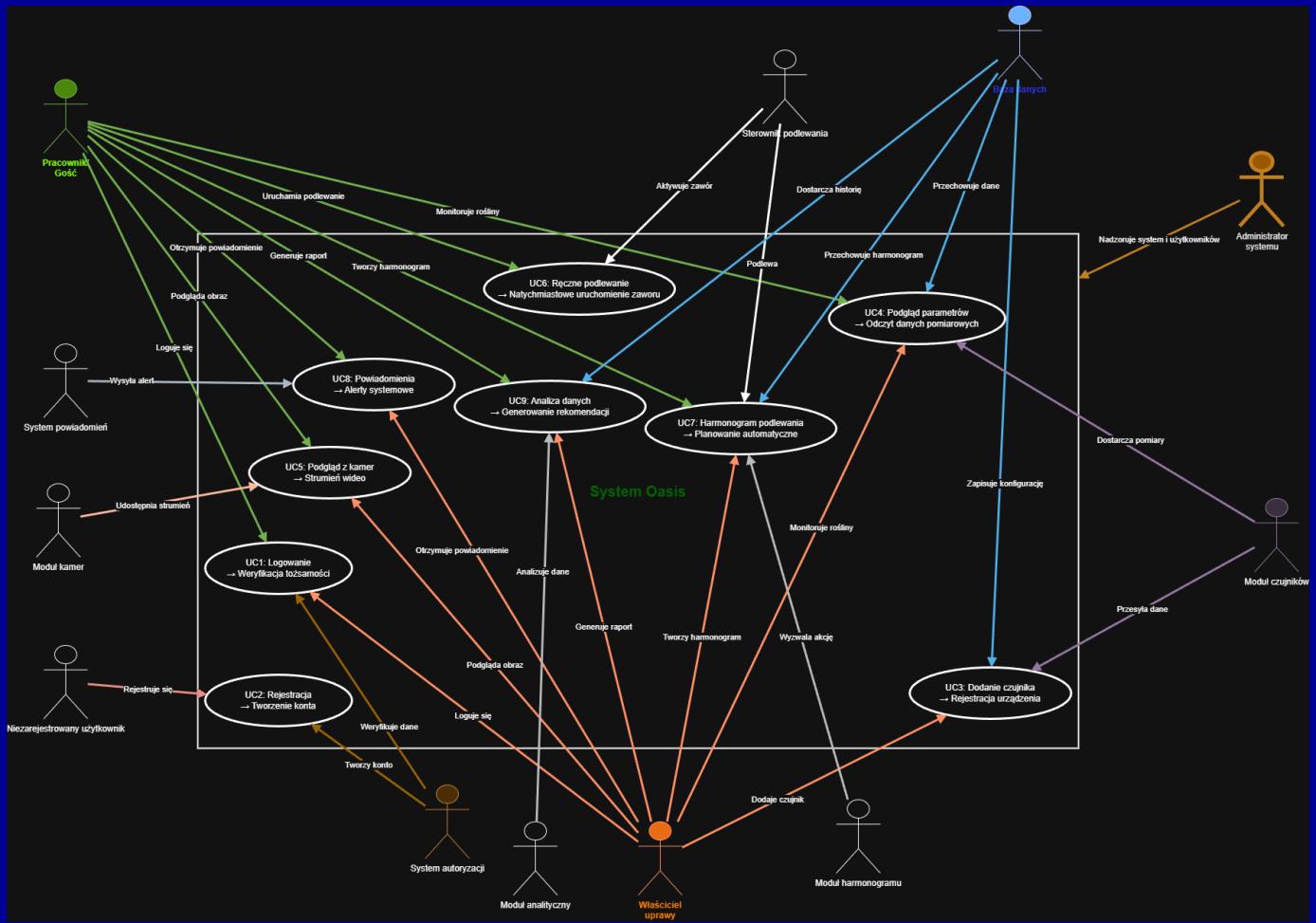
Baza danych	Przechowuje parametry roślin, dane historyczne, konfiguracje czujników, harmonogramy i logi zdarzeń.	Struktura danych
Moduł czujników	Urządzenia pomiarowe (wilgotność gleby, temperatura, ciśnienie, nasłonecznienie).	Urządzenie/komponent
Moduł kamer	Urządzenia wideo umożliwiające podgląd na żywo.	Urządzenie/komponent
Sterownik podlewania	Fizycznie uruchamia zawory i pompy wody.	Urządzenie/komponent
Moduł harmonogramu	Planowanie i automatyczne wyzwalanie akcji podlewania.	Usługa/system
System powiadomień	Wysyła alerty (np. niski poziom wilgotności, awaria czujnika).	Usługa/system
System autoryzacji	Weryfikuje tożsamość użytkowników i zarządza dostępem.	Usługa/system
Moduł analityczny	Analizuje dane historyczne i generuje raporty/rekomendacje.	Usługa/system

Przypadki użycia (Use cases)

Nr	Nazwa	Aktorzy	Scenariusz główny	Rozszerzenia
UC1	Logowanie do systemu	Zarejestrowany użytkownik, System autoryzacji	1. Użytkownik podaje login i hasło. 2. System autoryzacji weryfikuje dane. 3. Użytkownik uzyskuje dostęp.	Błędne dane → komunikat o błędzie. Konto zablokowane → brak dostępu.
UC2	Rejestracja użytkownika	Niezarejestrowany użytkownik, System autoryzacji	1. Użytkownik podaje dane rejestracyjne. 2. System zapisuje konto. 3. Użytkownik może się zalogować.	Brak wymaganych danych → komunikat o błędzie.
UC3	Dodanie czujnika	Właściciel uprawy, Moduł czujników, Baza danych	1. Użytkownik wybiera „Dodaj czujnik”. 2. Podaje parametry. 3. System zapisuje w bazie. 4. Czujnik raportuje dane.	Brak połączenia z czujnikiem → komunikat o błędzie.
UC4	Podgląd parametrów roślin	Zarejestrowany użytkownik, Moduł czujników, Baza danych	1. Użytkownik wybiera region. 2. System pobiera dane. 3. Wyświetla parametry.	Brak danych → komunikat o błędzie.
UC5	Podgląd obrazu z kamer	Zarejestrowany użytkownik, Moduł kamer	1. Użytkownik wybiera kamerę. 2. System zestawia strumień. 3. Obraz wyświetlany w aplikacji.	Kamera offline → komunikat o błędzie.

UC6	Ręczne podlewanie	Zarejestrowany użytkownik, Sterownik podlewania	1. Użytkownik wybiera „Podlej teraz”. 2. System wysyła sygnał. 3. Sterownik uruchamia zawór/pompę.	Brak wody/awaria → komunikat o błędzie.
UC7	Harmonogram podlewania	Zarejestrowany użytkownik, Moduł harmonogramu, Sterownik podlewania, Baza danych	1. Użytkownik ustawia harmonogram. 2. System zapisuje w bazie. 3. Moduł wyzwała akcję. 4. Sterownik uruchamia podlewanie.	Konflikt harmonogramów → komunikat i propozycja korekty.
UC8	Powiadomienia o zdarzeniach	System powiadomień, Zarejestrowany użytkownik	1. System wykrywa zdarzenie. 2. Generuje powiadomienie. 3. Wysyła do użytkownika.	Brak łączności → powiadomienie zapisane w kolejce.
UC9	Analiza danych i raporty	Zarejestrowany użytkownik, Moduł analityczny, Baza danych	1. Użytkownik wybiera „Raport”. 2. System pobiera dane historyczne. 3. Generuje wykresy i rekomendacje.	Brak danych → komunikat o braku raportu.

Diagram Use Case:



3. Specyfikacja wymagań

3.1 Wymagania funkcjonalne

- [F(unctional)-01] System umożliwia rejestrację, logowanie i wylogowanie użytkownika.
- [F-02] System autoryzacji weryfikuje dane logowania.
- [F-03] Dostęp do funkcji systemu mają wyłącznie zalogowani użytkownicy.
- [F-04] Użytkownik może dodawać czujniki i przypisywać je do regionów.
- [F-05] Użytkownik może dodawać kamery i przypisywać je do regionów.

- [F-06] System zapisuje konfigurację czujników i kamer w bazie danych.
- [F-07] System prezentuje aktualne dane z czujników.
- [F-08] System umożliwia podgląd obrazu z kamer w czasie rzeczywistym.
- [F-09] System przechowuje dane historyczne w bazie danych.
- [F-10] Użytkownik może uruchomić podlewanie ręcznie.
- [F-11] Użytkownik może ustawić harmonogram podlewania.
- [F-12] Moduł harmonogramu automatycznie wyzwala podlewanie zgodnie z planem.
- [F-13] Sterownik podlewania realizuje fizyczne otwarcie/zamknięcie zaworów.
- [F-14] System wysyła powiadomienia o zdarzeniach.
- [F-15] Powiadomienia mogą być dostarczane jako e-mail, SMS lub push.
- [F-16] System generuje raporty z danych historycznych.
- [F-17] Moduł analityczny sugeruje optymalne czasy podlewania.

3.2 Wymagania pozafunkcjonalne

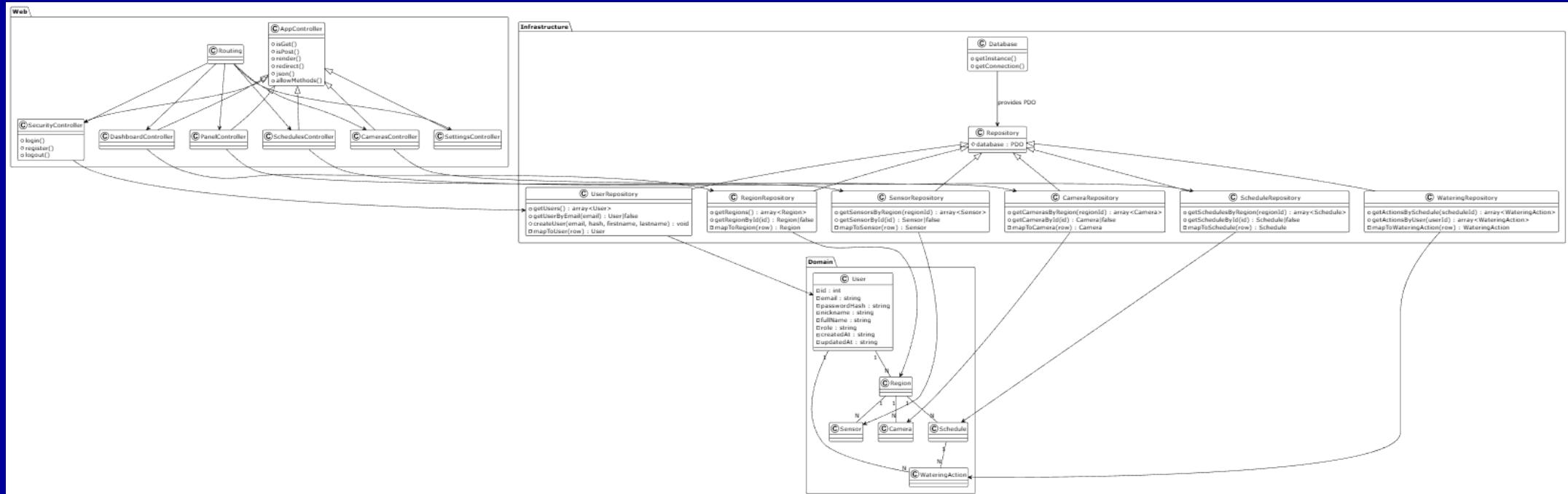
- [N(on)F(functional)-01] Bezpieczeństwo: dane muszą być szyfrowane (HTTPS, hasła hash).
- [NF-02] Dostępność: system dostępny $\geq 99\%$ czasu w tygodniu.
- [NF-03] Skalowalność: możliwość dodania wielu czujników i kamer bez utraty wydajności.
- [NF-04] Wydajność: aktualizacja danych czujników $\leq 5\text{ s}$.
- [NF-05] Użyteczność: intuicyjny interfejs, wsparcie dla urządzeń mobilnych i desktopowych (responsywność aplikacji).
- [NF-06] Rozszerzalność: integracja z zewnętrznymi systemami IoT przez API.

Wybór technologii

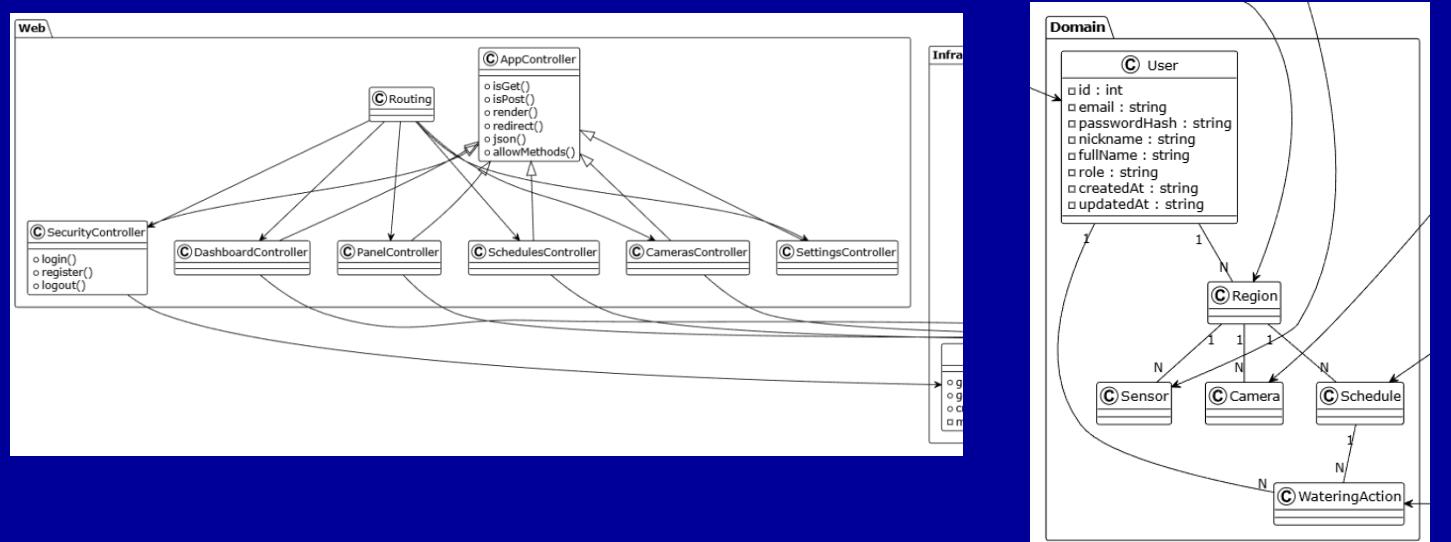
Na podstawie zdefiniowanych założeń, optymalnym wyborem staje się projekt wykorzystujący technologię języków: PHP, JavaScript, HTML/CSS oraz podział aplikacji na warstwy monolitu MVC: Models, Views, Controllers.

Poniżej znajduje się orientacyjny diagram UML prezentujący wstępную strukturę projektu.

Diagram UML



Dzieląc na mniejsze sekcje:



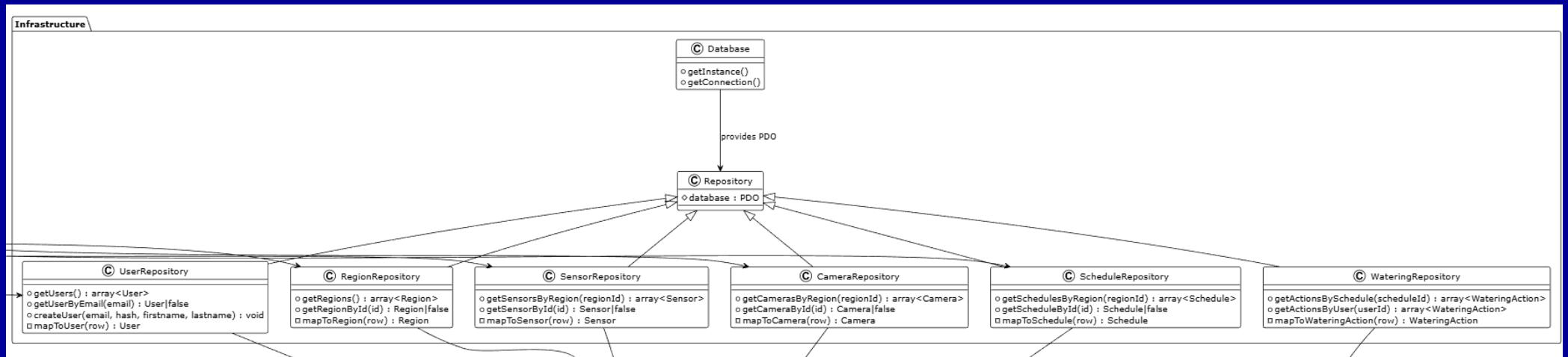
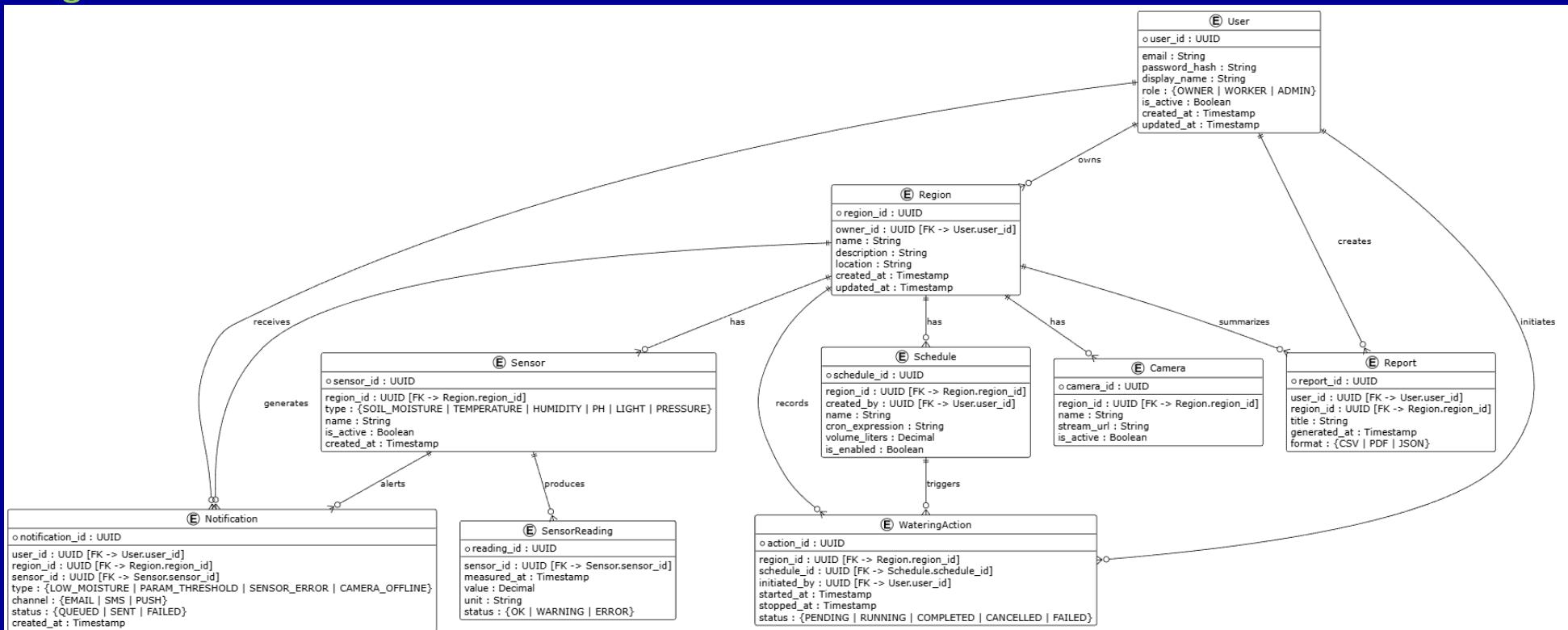


Diagram ERD



Podsumowanie

Powyższa dokumentacja skupia się na ogólnym rozrysowaniu funkcjonowania projektu aplikacji. Oczywiście ostateczna wersja produktu może się miejscami znacznie różnić od założeń w wymaganiach i diagramach. Może to być wynik nieprzewidzianych luk w architekturze, braków wiedzy lub niewystarczających zasobów czasu do realizacji projektu zadanym czasie.