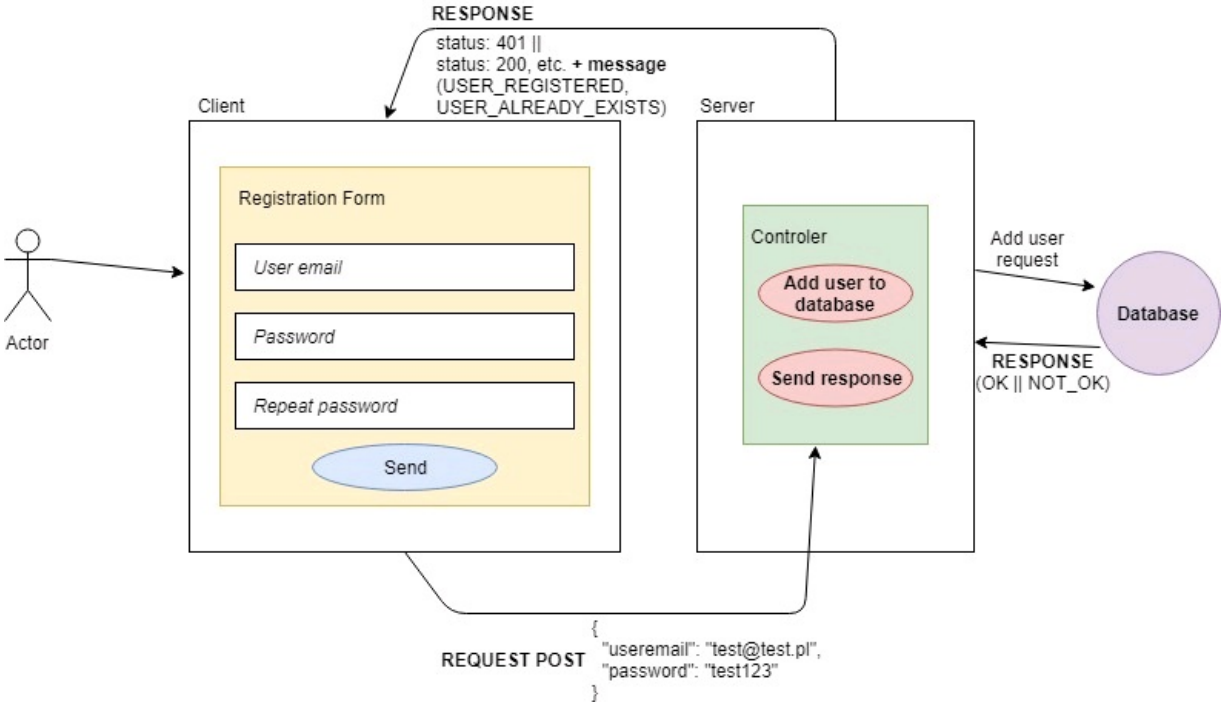
Ogólny przypadek użycia

1. Użytkownik zakłada konto za pośrednictwem klienta WWW

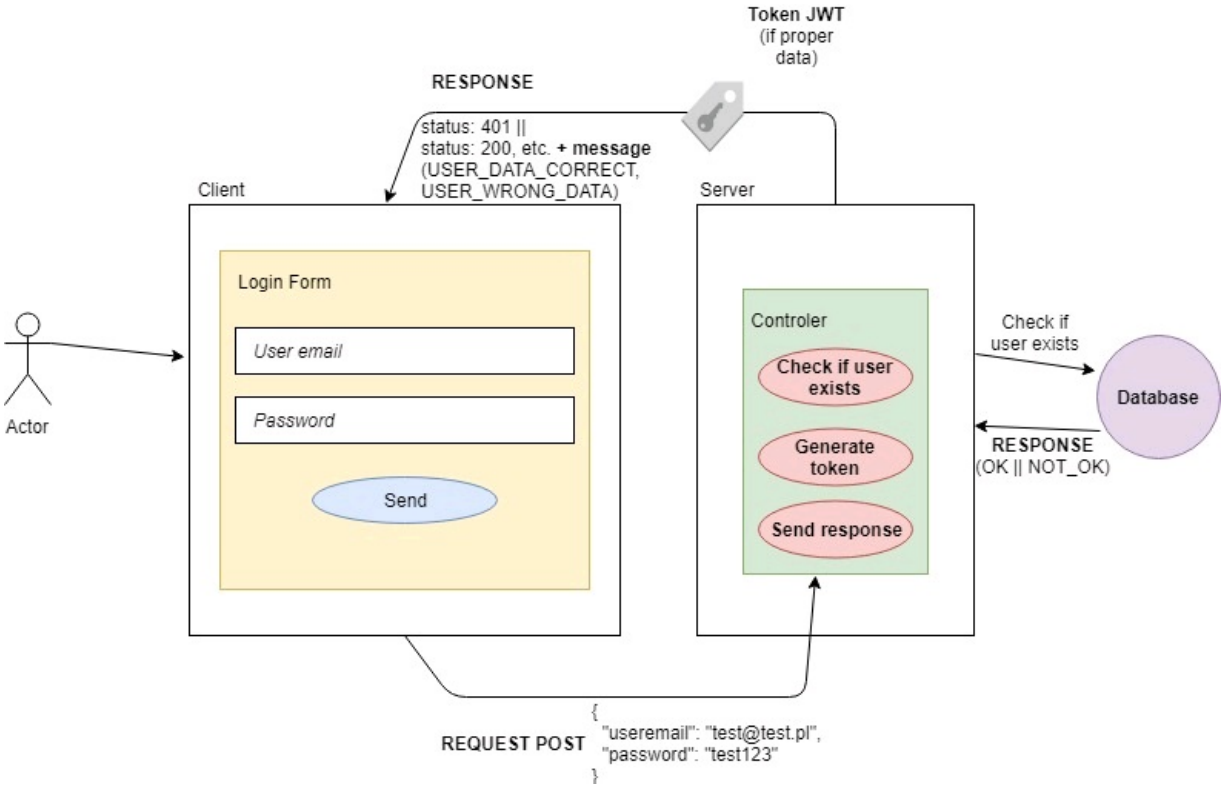


Przykładowy model użytkownika:

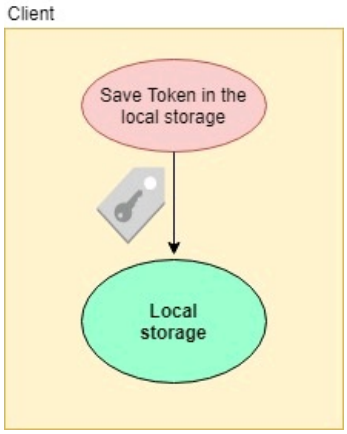
```
{
  "useremail": "test@test.pl",
  "salt": "9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08",
  "hash": "4417a30dc8c6b53f5e2e2b90511593480360l7f9061e8ace1f966a1ab58fbedd"
}
```

2. Użytkownik loguje się do systemu

W momencie wysłania formularza, klient kieruje do serwera zapytanie POST, zawierające dane użytkownika - adres mailowy oraz hasło. Serwer kontaktuje się z bazą danych, w celu zweryfikowania danych. Jeśli są one prawidłowe, generuje token JWT, który następnie zwraca do klienta. Od tej pory klient może potwierdzać swoją tożsamość przed serwerem. Token musi być dostarczony jako nagłówek każdego requestu płynącego od klienta do serwera, proszącego o zasoby, które powinny być zabezpieczone. Na podstawie tokenu, serwer będzie również w stanie rozpoznać jaki użytkownik się nim posługuje (w tokenie zakodowany jest posługujący się nim użytkownik). Dzięki temu zawsze zwróci zasoby, które należą do konkretnego użytkownika.

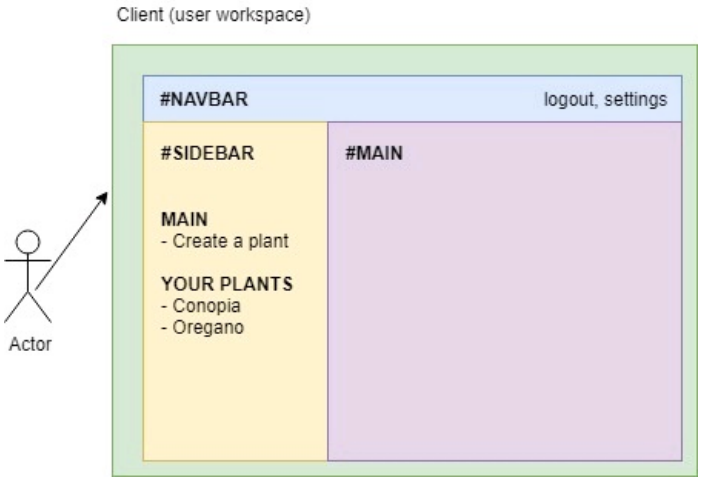


Klient zapisuje token w swojej pamięci lokalnej, dzięki temu jest w stanie posługiwać się nim zawsze, gdy zajdzie taka potrzeba.



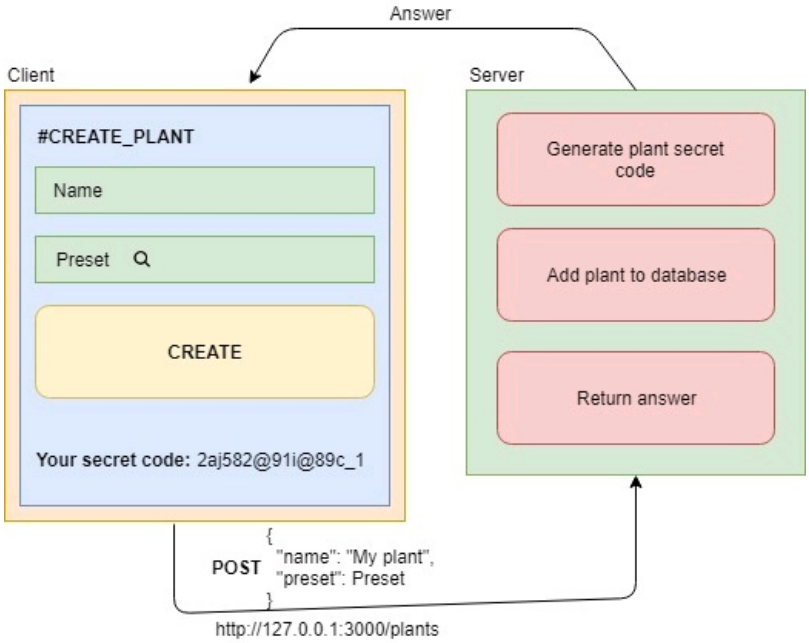
3. Przestrzeń robocza użytkownika

Przestrzeń użytkownika jest bardzo prosta. Składa się na nią kilka mniejszych komponentów: * #NAVBAR - zawiera przycisk pozwalający na wylogowanie oraz przejście do ustawień użytkownika * #SIDEBAR - zawiera główne akcje, które może wykonać użytkownik (Create a plan) oraz aktualne uprawy użytkownika. * #MAIN - główna część przestrzeni roboczej



4. Użytkownik tworzy plantację

Za pośrednictwem formularza, użytkownik może stworzyć nową plantację oraz przypisać do niej preset.



Przykładowy preset:

```
{
  "name": "Canabis",
  "goodTemperature": 23,
  "goodHumidity": 70,
  "waterPerDay": 1,
  "howManyTimesToWater": 3,
  "expectedGrowth": 2
}
```

Preset zawiera podstawowe parametry dla konkretnej uprawy. Posłużą one do stworzenia prognozy wzrostu rośliny oraz kontroli parametrów panujących na uprawie. Preset powinien mieć również określony możliwe odchylenia dla poszczególnych parametrów (takie, bez wystąpienia których nie będzie generowany alarm).

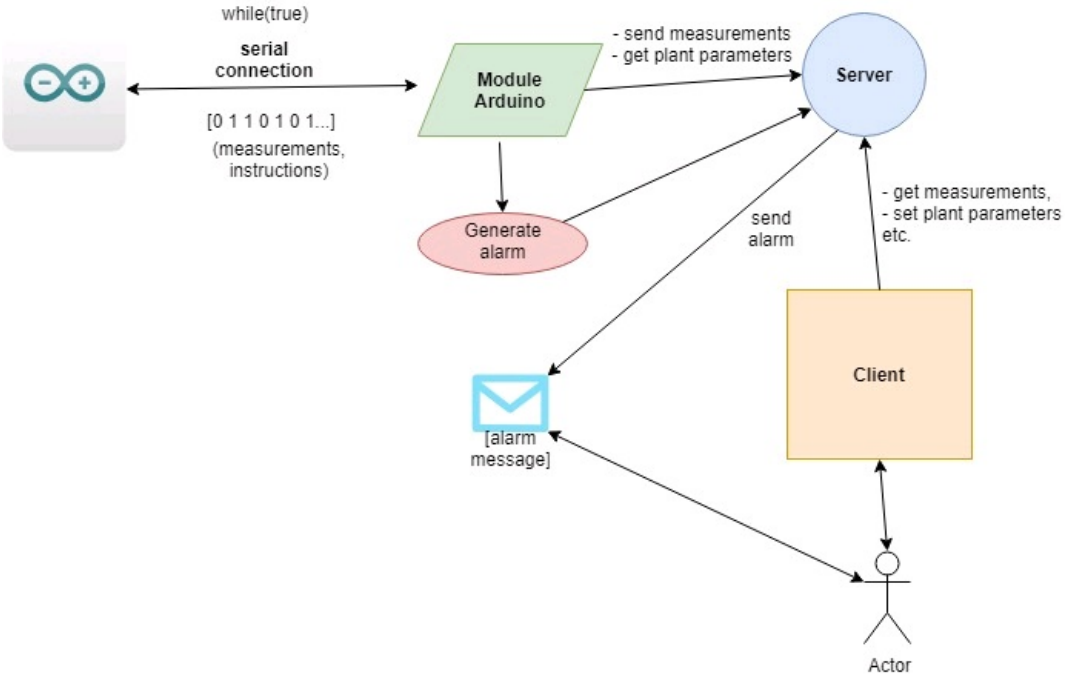
Użytkownik otrzymuje również od serwera tajny kod plantacji, dzięki któremu będzie mógł skonfigurować swój moduł Arduino.

5. Konfiguracja modułu Arduino

Użytkownik wprowadza tajny klucz uprawy, do pliku konfiguracyjnego:

```
plantSecretKey: '2aj582@91i@89c_1'
```

6. Bardziej szczegółowy schemt komunikacji



7. Podłączenie płytyki Arduino

Płytka zostaje podłączona do modułu Arduino - w celach testowych przyjmijmy, że za pośrednictwem klasycznego łącza szeregowego (otwarte gniazdo, Arduino działa w pętli). Wykorzystanie zostanie przykładowo moduł PySerial (pozwala on na obsługę szeregowego połączenia przy pomocy Pythona).

8. Inicjalizacja modułu Arduino

Na początku moduł wysyła do serwera komunikat READY, zawierający tajny kod uprawy. Serwer w ramach odpowiedzi zwraca do modułu parametry, które powinny być przestrzegane na plantacji. Moduł zapisuje te parametry w swojej pamięci.

Od tego momentu rozpoczyna się praca modułu Arduino (w nieskończonej pętli).

9. Praca Moduł Arudino - Serwer

Moduł zwraca się do serwera (w nieskończonej pętli) sprawdzając czy: * zmieniły się ustalone parametry plantacji (min. czas co jaki serwer oczekuje od modułu pomiarów) * powinien dodać do listy swoich czynności uruchamianie spryskiwacza codziennie o X godzinie

Na podstawie otrzymanych informacji odpowiednio reaguje (np. o danej godzinie przekazuje do płytki instrukcje URUCHOM_SPRYSKIWACZE lub zmniejsza uśpienie wątku pobiarającego z płytki pomiary).

Moduł co określony czas X pobiera z płytki (za pośrednictwem portu szeregowego) pomiary i przesyła je do serwera w zapytaniu (wraz z kodem uprawy).

Przykładowe zapytanie:

```
-> http://127.0.0.1:3000/measurements
POST:
{
  'plantSecretCode': '2aj582@91i@89c_1',
  'time': '2019-03-15 21:44:11',
  'temperature': 21,
  'humidity': 68
}
```

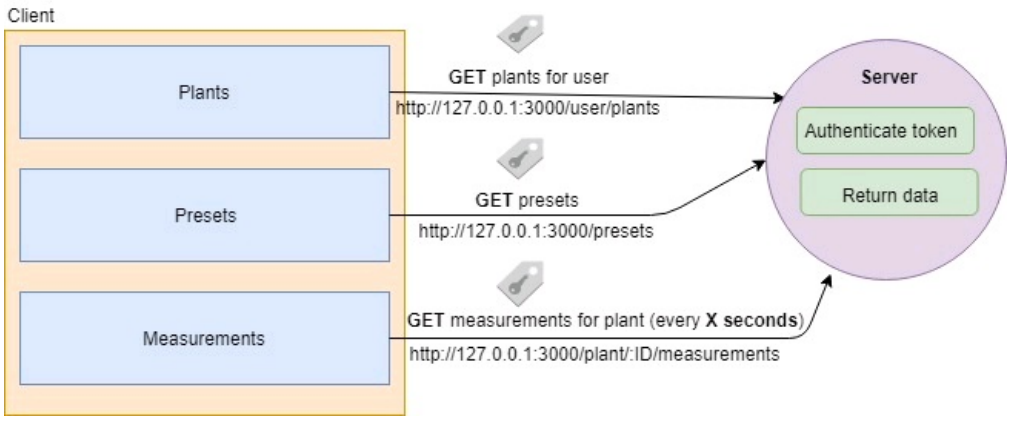
Oczywiście jeśli uprawa o takim tajnym kodzie nie istnieje w bazie danych, serwer powinien odrzucić taki pomiar (a nawet zablokować zapytania od danego adresu IP [bezpieczeństwo]). Po otrzymaniu pomiarów od prawidłowej uprawy, serwer zapisuje je w bazie danych (oczywiście referują one do konkretnej uprawy znanej dzięki tajnemu kodowi).

UWAGA: Serwer jedynie zwraca odpowiedzi, ale sam nie zwraca się do modułu Arduino z prośbą o cokolwiek. UWAGA 2: Moduł arduino może realizować funkcję alarmowania. Wówczas wstępnie analizował by pomiary, które otrzymuje z płytki i porównywał ze wskaźnikami (presetem) obowiązującym na uprawie. Jeżeli coś by się nie zgadzało, zgłaszałby to serwerowi, który następnie wysyłał by wiadomość do

właściciela plantacji (np. na adres mailowy). [alarmowanie_1]

10.. Praca Klient - Serwer - Moduł arduino

Klient po inicjalizacji (otworzeniu strony przez użytkownika i zalogowaniu się do systemu) pobiera z serwera podstawowe dane: * dostępne uprawy * dostępne presety * aktualne pomiary dla upraw



Oczywiście klient do każdego żądania dołącza token JWT, dzięki czemu serwer może zwrócić zasoby tylko ich prawowitemu właścicielowi.

Klient będzie, między innymi: * prezentował historię pomiarów na uprawie * prezentował aktualny stan uprawy * prezentował prognozy dla uprawy * dawał możliwość zmiany warunków uprawy (presetu) * dawał możliwość definiowania nowych presetów * dawał możliwość dodawania nowych akcji (np. podlewanie) do bazy

Nie będzie on jednak bezpośrednio komunikował się z modułem Arduino. Wykorzystywał będzie jedynie dane zapisane przez serwer w bazie danych oraz dodawał własne (istotne dla użytkownika oraz związane z konkretną uprawą) dane do bazy. Pamiętajmy, że moduł Arduino stale pracuje (w nieskończonej pętli) i zapisuje dane do bazy (za pośrednictwem serwera) oraz pobiera dane z bazy (również za pośrednictwem serwera). Dzięki temu klient, mający dostęp do bazy danych (za pośrednictwem serwera) posiada aktualne dane pomiarowe oraz ma pewność, że dodane przez niego do bazy dane (również za pośrednictwem serwera), zostaną odpowiednio zinterpretowane przez moduł Arduino.

Najistotniejsza jednak kwestia to kiedy klient powinien odpytywać serwer o aktualną sytuację na konkretnej uprawie. Istnieje kilka opcji: * wprowadzenie przycisku odśwież * odpytywanie co ustalony dla uprawy czas X + kilka dodatkowych sekund (uwzględnienie opóźnienia) * odpytywanie co każdorazowe odświeżenie strony * odpytywanie w momencie każdorazowego przeładowania widoku

Gdybyśmy chcieli dodatkowo realizować alarmowanie po stronie klienta, to musiałby on być w stanie ciągłej gotowości [alarmowanie_2]. Reagowanie na przekroczenie wskaźników możemy rozwiązać więc po stronie modułu Arduino [alarmowanie_1], dając klientowi możliwość oceny parametrów uprawy w momencie odświeżenia jej (np. zmiana nazwy parametru na czerwony, wyświetlenie komunikatu). Pierwsza linia alarmowania (pierwsza linia ognia) działać będzie jednak po stronie modułu Arduino oraz serwera.

[alarmowanie_2]: Klient posiada uruchomioną pętlę, odpalaną co czas X (dla konkretnej uprawy) + kilka dodatkowych sekund. W momencie uruchomienia pętli, pobiera od serwera parametry panujące na plantacji i reaguje (np. wysyła alarm).

11. [alarmowanie_1] vs [alarmowanie_2]

Pierwszy sposób wydaje się mieć przewagę nad drugim, ponieważ umożliwia rozdzielenie głównych funkcji poszczególnych modułów: * moduł Arduino pobiera pomiary z płytki oraz alarmuje użytkownika (pierwsza linia ognia) * klient umożliwia użytkownikowi przeglądanie stanu plantacji (spokojna głowa), dostęp do historii (kontrola), określania parametrów plantacji (po raz kolejny kontrola) oraz przeglądania statystyk (wiedza)

Drugi sposób posiada również jedną wadę: wymaga synchronizacji pomiędzy modułem Arduino a klientem. Nie chcemy wprowadzać dodatkowej zależności.