



# BREWIT

Dokumentacja projektowa PZSP2

WERSJA 1

10.01.2025

Semestr 2024Z

Zespół nr 12 w składzie:

Michał Goławski, Angelika Ostrowska, Marcin Sikorski, Maciej Dobrowolski

Mentor zespołu: Krzysztof Cabaj

Właściciel tematu: Piotr Pałka

## Spis treści

1	Wprowadzenie .....	3
1.1	Cel projektu .....	3
1.2	Wstępna wizja projektu .....	3
2	Metodologia wytwarzania .....	4
2.1	Komunikacja .....	4
2.2	Dobre praktyki pracy w zespole .....	4
2.3	Role w zespole .....	5
3	Analiza wymagań .....	6
3.1	Wymagania użytkownika i biznesowe .....	6
3.1.1	Wymagania biznesowe .....	6
3.1.2	Wymagania użytkownika .....	6
3.2	Wymagania funkcjonalne i нефункционалне .....	8
3.3	Przypadki użycia .....	11
3.3.1	Biznesowe przypadki użycia .....	11
3.3.2	Systemowe przypadki użycia .....	15
3.3.3	Diagramy przypadków użycia .....	26
3.4	Potwierdzenie zgodności wymagań .....	27
4	Definicja architektury .....	28
4.1	Plan struktury systemu .....	28
4.1.1	Diagram systemów .....	28
4.1.2	Diagram kontenerów .....	29
4.1.3	Diagram komponentów .....	30
4.2	Szablony architektoniczne .....	34
4.3	Wdrożenie .....	35
5	Dane trwałe .....	36
5.1	Model logiczny danych .....	36
5.2	Przetwarzanie i przechowywanie danych .....	38
6	Specyfikacja analityczna i projektowa .....	39
6.1	Metoda realizacji .....	39
6.2	Diagram klas modelu danych .....	39
6.3	Statystyki .....	40
7	Projekt standardu interfejsu użytkownika .....	41

8	Specyfikacja testów.....	48
8.1	Rodzaje testów.....	48
8.2	Miara jakości testów:.....	48
9	<i>Wirtualizacja/konteneryzacja</i> .....	49
9.1	Konfiguracja Docker Compose.....	49
9.1.1	Reverse Proxy (NGINX).....	49
9.1.2	Frontend.....	49
9.1.3	Backend.....	49
9.1.4	Baza danych .....	49
10	Bezpieczeństwo.....	50
11	Podręcznik użytkownika.....	50
11.1	Administrator .....	50
11.2	Browar kontraktowy .....	50
11.3	Browar produkcyjny.....	50
12	Podręcznik administratora.....	51
12.1	Instrukcja uruchomienia systemu.....	51
12.2	Instrukcja uruchomienia komponentów.....	51
12.2.1	Baza danych .....	51
12.2.2	Backend.....	51
12.2.3	Frontend.....	52
12.3	Zarządzanie użytkownikami i uprawnieniami.....	52
13	Podsumowanie.....	52
13.1	Realizacja wymagań .....	52
13.1.1	Wymagania funkcjonalne.....	52
13.1.2	Wymagania нефункционалне.....	52
13.2	Mocne strony systemu.....	52
13.3	Słabe strony systemu .....	53
13.4	Możliwy rozwój .....	53
14	Bibliografia .....	53

# 1 Wprowadzenie

## 1.1 Cel projektu

Problem: Trudność w komunikacji między browarami kontraktowymi a komercyjnymi oraz w zarządzaniu wynajmowaniem sprzętu spowodowana brakiem rozwiązania dostosowanego do szczególnych cech tego sektora.

Cel: Stworzenie aplikacji, która będzie wspomagała kontrakty między wieloma browarami kontraktowymi i wieloma browarami komercyjnymi.

## 1.2 Wstępna wizja projektu

Aplikacja będzie miała 3 rodzaje użytkowników: administrator, browary komercyjne, browary kontraktowe.

Administrator będzie miał dostęp do zagregowanych danych z wybranego przez siebie czasu, będzie mógł sprawdzić, ile piwa zostało wyprodukowane, ile jest browarów komercyjnych, ile jest browarów kontraktowych, historię średniej 'zajętości sprzętu' browarów komercyjnych.

Pracownik browaru komercyjnego wprowadza do aplikacji informacje o możliwych do udostępnienia zasobach - ich ilość, parametry, jakie rodzaje piwa browar komercyjny jest w stanie wytworzyć, jakich nie akceptuje. Pracownik może też sprawdzić jakie osoby powinny mieć dostęp do sprzętu na podstawie rezerwacji w systemie.

Pracownik browaru kontraktowego ma wgląd w ofertę wielu browarów komercyjnych i może zarezerwować na dany okres zasoby browaru, który spełnia jego oczekiwania. Podczas produkcji piwa, browar kontraktowy musi na bieżąco aktualizować wykonane kroki receptury w aplikacji. Dzięki temu po skończonej produkcji cały proces jest udokumentowany w systemie.

## 2 Metodologia wytwarzania

### 2.1 Komunikacja

Komunikacja odbywa się osobiście oraz za pomocą czatu grupowego na platformie Discord. Wymieniane są tam bieżące informacje dotyczące projektu. Organizowane są tam również spotkania głosowe.

Pracę nad projektem wykonujemy m.in. na spotkaniach zdalnych całego zespołu. Są one regularne, tematyka konieczna do poruszenia na kolejnym spotkaniu ustalana jest na spotkaniu poprzednim. Terminy spotkań ustalane są poprzez porównanie dostępności wszystkich członków. W przypadku konfliktów wybierany jest termin pasujący największej liczbie osób.

W przypadku konfliktów i wątpliwości pojawiających się podczas pracy nad projektem, odbywa się wymiana punktów widzenia, rozważenie plusów i minusów obu rozwiązań i wspólne ustalenie na którą z opcji się decydujemy.

Do wstępnego planowania dokumentacji korzystamy z narzędzia Google Docs oraz draw.io, po czym zmiany przenoszone są do dokumentu znajdującego się na platformie Teams.

### 2.2 Dobre praktyki pracy w zespole

Stosujemy elementy programowania zwinnego (ang. Agile) ze względu na konieczność przedstawienia wyników w prezentacji śródsesemestralnej w celu usprawnienia pracy w zespole.

Wartości Agile na które szczególnie zwracamy uwagę to:

- dostarczanie gotowego oprogramowania z kilkutygodniowym odstępem,
- gotowość na zmiany wymagań w trakcie powstawania oprogramowania (ze względu na konieczność konsultacji z ekspertami),
- tworzenie projektów wokół zmotywowanych ludzi,
- najlepsze rozwiązania architektoniczne, wymagania i projekty pochodzą od samoorganizujących się zespołów,
- najbardziej efektywnym i wydajnym sposobem przekazywania informacji zespołowi deweloperskiemu i wewnątrz niego jest rozmowa twarzą w twarz (z tego powodu często spotykamy się osobiście i na rozmowy zdalne).

Projekt będzie używał Continuous Integration do regularnego sprawdzania i testowania nowych wersji aplikacji.

## 2.3 Role w zespole

### A. W ujęciu Belbina:

Każdy z członków zespołu wykonał test Belbina. Jego wyniki to:

Maciej Dobrowolski – Specialist, Team worker, Completer, Evaluator

Michał Goławski – Plant, Completer, Shaper

Marcin Sikorski – Team worker, Completer, Shaper, Resource investigator

Angelika Ostrowska – Plant, Team worker, Specialist

Zatem nasz zespół składa się z:

3 Completer

2 Shaper

0 Implementer

1 Evaluator

2 Plant

2 Specialist

0 Coordinator

1 Resource investigator

3 Team worker

Można zauważyć, że w zespole brakuje koordynatora i implementatorów, więc konieczna będzie samoorganizacja zespołu i wzajemnie motywowanie do pracy.

### B. Merytoryczne:

Frontend - Angelika Ostrowska

Backend - Michał Goławski, Maciej Dobrowolski

Bazy danych - Marcin Sikorski

## 3 Analiza wymagań

### 3.1 Wymagania użytkownika i biznesowe

#### 3.1.1 Wymagania biznesowe

**B1.** Browary komercyjne potrzebują systemu umożliwiającego wynajem swoich niewykorzystanych zasobów w celu generowania dodatkowego przychodu.

**B2.** Browary kontraktowe potrzebują systemu ułatwiającego wyszukiwanie browaru komercyjnego spełniającego ich wymagania.

**B3.** Browary kontraktowe potrzebują systemu ułatwiającego przechowywanie postępów realizacji receptury.

**B4.** Browary komercyjne potrzebują systemu, który ułatwi ich promocję wśród potencjalnie zainteresowanych browarów kontraktowych.

#### 3.1.2 Wymagania użytkownika

##### 3.1.2.1 Browar komercyjny

**U1.** Pracownik browaru komercyjnego może wysłać prośbę o rejestrację podając podstawowe dane firmy i dane do logowania.

**U2.** Pracownik browaru komercyjnego może wprowadzić do systemu informacje o dostępnych do udostępnienia urządzeniach.

**U3.** Pracownik browaru komercyjnego może sprecyzować parametry urządzeń, takie jak możliwości produkcyjne (wybicie, warka), technologie rozlewu, osiągalna temperatura.

**U4.** Pracownik browaru komercyjnego może sprawdzić harmonogram rezerwacji urządzeń.

**U5.** Pracownik browaru komercyjnego w harmonogramie widzi informacje o rodzaju urządzenia, czasie rezerwacji, browarze kontraktowym który ją rezerwuje i rodzaju piwa w niej produkowanego.

**U6.** Pracownik browaru komercyjnego może sprawdzić jakie osoby mają wstęp na teren w danym dniu.

**U7.** Pracownik browaru komercyjnego może określić cennik za wykorzystywanie danych urządzeń na dany okres.

**U8.** Pracownik browaru komercyjnego może przyjąć lub odrzucić prośbę o rezerwację urządzeń.

**U9.** Pracownik browaru komercyjnego może anulować aktualną rezerwację swojego sprzętu.

**U10.** Pracownik browaru komercyjnego może przeglądać swoje aktywne rezerwacje.

**U11.** Pracownik browaru komercyjnego może przeglądać przeszłe rezerwacje z ich końcowym rezultatem.

### 3.1.2.2 Browar kontraktowy

**U12.** Pracownik browaru kontraktowego może przeglądać swoje aktywne rezerwacje.

**U13.** Pracownik browaru kontraktowego może wysłać prośbę o rejestrację podając podstawowe dane firmy i dane do logowania.

**U14.** Pracownik browaru kontraktowego może sprawdzić ofertę różnych browarów komercyjnych.

**U15.** Pracownik browaru kontraktowego może filtrować oferty po różnych parametrach (rodzaju, dostępności w czasie, pojemności, wybiciu, zakresie PH wody, zakresie temperatury, rodzajach rozlewu) w kolejnych etapach produkcji.

**U16.** Pracownik browaru kontraktowego może sprawdzić harmonogram dostępności urządzeń.

**U17.** Pracownik browaru kontraktowego może sprawdzić parametry urządzeń.

**U18.** Pracownik browaru kontraktowego może sprawdzić dzienną cenę za wypożyczenie urządzenia.

**U19.** Pracownik browaru kontraktowego może złożyć prośbę o zarezerwowanie danych urządzenia w danym browarze komercyjnym na dany okres.

**U20.** Pracownik browaru kontraktowego może określić, ile piwa zamierza wyprodukować.

**U21.** Pracownik browaru kontraktowego może podać listę osób, które będą w danym czasie przebywać na terenie browaru komercyjnego.

**U22.** Pracownik browaru kontraktowego może zadeklarować konieczność braku sąsiedztwa innych browarów w trakcie wytwarzania piwa w celu zachowania tajemnicy receptury (w danym sektorze).

**U23.** Pracownik browaru kontraktowego może sprawdzić całkowitą kwotę należną za skorzystanie z usług browaru komercyjnego.

**U24.** Pracownik browaru kontraktowego może wprowadzić recepturę produkcji danego rodzaju piwa.

**U25.** Pracownik browaru kontraktowego może anulować swoją rezerwację.

**U26.** Pracownik browaru kontraktowego może uzupełniać historię wykonania danej partii piwa (dokładny czas i szczegóły wykonania etapu)).

**U27.** Pracownik browaru kontraktowego może zapisać czy dana partia piwa została wykonana z sukcesem.

**U28.** Pracownik browaru kontraktowego może przeglądać przeszłe rezerwacje wraz z ich końcowym rezultatem i historią wykonywania.

**U29.** Pracownik browaru kontraktowego może przeglądać receptury swojego browaru i nimi zarządzać (dodawać, usuwać, edytować).



### 3.1.2.3 Administrator

**U30.** Administrator systemu może akceptować i odrzucać prośby o rejestrację browarów komercyjnych i kontraktowych.

**U31.** Administrator systemu może sprawdzić, ile browarów komercyjnych i kontraktowych jest zarejestrowanych w systemie.

**U32.** Administrator systemu może zobaczyć listę browarów komercyjnych i kontraktowych jest zarejestrowanych w systemie.

**U33.** Administrator systemu może sprawdzić, ile sumarycznie piwa produkowane jest we wszystkich browarach komercyjnych.

## 3.2 Wymagania funkcjonalne i нефункционалне

Funkcjonalne:

- Aplikacja powinna pozwalać na wysłanie prośby o zarejestrowanie browaru po podaniu podstawowych danych firmy i danych do logowania.
- Aplikacja powinna pozwalać na zalogowanie się.
- Aplikacja powinna pozwalać administratorowi na zarządzanie prośbami o rejestrację konta.
- Aplikacja powinna umożliwiać zobaczenie współczynnika nieudanych procesów wytwarzania piwa w browarach komercyjnych.
- Aplikacja powinna pozwalać browarom komercyjnym na zarządzanie (dodawanie, usuwanie, modyfikacja) swoimi urządzeniami.
- Aplikacja powinna pozwalać browarom komercyjnym na sprecyzowanie parametrów urządzeń, takich jak pojemność, wybicie, technologie rozlewu, temperatura fermentacji/leżakowania itp.
- Aplikacja powinna pozwalać browarom komercyjnym na ustalenie cennika za wykorzystywanie danych urządzeń na dany okres.
- Aplikacja powinna pozwalać browarom komercyjnym na sprawdzenie w harmonogramie rezerwacji danego urządzenia informacji o czasie rezerwacji, browarze kontraktowym który ją rezerwuje i rodzaju piwa w niej produkowanego.
- Aplikacja powinna pozwalać browarom komercyjnym i kontraktowym na przeglądanie i anulowanie aktualnych rezerwacji.
- Aplikacja powinna pozwalać browarom komercyjnym na sprawdzenie jakie osoby mają wstęp na teren browaru w danym dniu.
- Aplikacja powinna pozwalać browarom komercyjnym na przyjęcie lub odrzucenie prośby o rezerwację urządzeń.
- Aplikacja powinna pozwalać browarom kontraktowym na przeglądanie przeszłych rezerwacji wraz z ich końcowym rezultatem i historią wykonywania.
- Aplikacja powinna pozwalać browarom kontraktowym na sprawdzenie oferty różnych browarów komercyjnych.

- Aplikacja powinna pozwalać browarom kontraktowym na filtrowanie urządzeń po ich rodzaju, dostępności w czasie, pojemności, wybiciu, zakresie PH wody, zakresie temperatury, rodzajach rozlewu, możliwości produkcji kwaśnego piwa, współczynnika nieudanych procesów wytwarzania piwa, współczynnika anulowanych rezerwacji.
- Aplikacja powinna pozwalać browarom kontraktowym na sprawdzenie harmonogramu dostępności urządzenia, parametrów urządzenia i ceny za wypożyczenie urządzenia na jeden dzień.
- Aplikacja powinna pozwalać browarom kontraktowym na złożenie prośby o zarezerwowanie urządzeń browaru komercyjnego na wybrany czas.
- Aplikacja powinna pozwalać browarom kontraktowym na planowanie produkcji piwa.
- Aplikacja powinna pozwalać browarom kontraktowym na określenie listy osób, które będą w danym czasie przebywać na terenie browaru komercyjnego.
- Aplikacja powinna pozwalać browarom kontraktowym na zadeklarować konieczność braku sąsiedztwa innych browarów w trakcie wytwarzania piwa w celu zachowania tajemnicy receptury.
- Aplikacja powinna pozwalać browarom kontraktowym na sprawdzenie całkowitej kwoty należnej za rezerwację.
- Aplikacja powinna pozwalać browarom kontraktowym na dodanie receptury produkcji danego rodzaju piwa.
- Aplikacja powinna pozwalać browarom kontraktowym na przeglądanie swoich receptur.
- Aplikacja powinna pozwalać browarom kontraktowym na uzupełnianie historii wykonania danej partii piwa (dokładny czas i szczegóły wykonania etapu).
- Aplikacja powinna pozwalać browarom kontraktowym na określenie czy dana partia piwa została wykonana z sukcesem.
- Aplikacja powinna umożliwić wyświetlanie współczynnika anulowanych rezerwacji przez dany browar.
- Aplikacja powinna umożliwić administratorowi wyświetlanie statystyki dotyczącej sumarycznej ilości piwa produkowanej przez wszystkie browary komercyjne.
- Aplikacja powinna umożliwić administratorowi wyświetlić listę oraz liczbę browarów komercyjnych i kontraktowych zarejestrowanych w systemie.

#### Niefunkcjonalne:

- Aplikacja powinna zapewniać zachowanie tajności receptur, poprzez zapewnienie, że tylko dany browar kontraktowy ma dostęp do dodanych przez siebie receptur.
- Aplikacja powinna zapewniać zachowanie tajności sposobów wykonania piwa, poprzez zapewnienie, że tylko dany browar kontraktowy ma dostęp do wprowadzonej historii wykonania danej partii.
- Aplikacja powinna zapewniać bezpieczeństwo dostępu do budynku browaru komercyjnego, poprzez udostępnianie listy osób, które danego dnia mogą w nim przebywać.
- Aplikacja powinna zapewniać, że niepowołany dostęp do danych części aplikacji będzie wykluczony.
- Aplikacja powinna zapewniać, że przy wyszukiwaniu ofert browarów komercyjnych serwer zwraca odpowiedź w czasie nie dłuższym niż 5 sekund zakładając, że z serwera korzysta naraz 1 osoba.
- Aplikacja powinna zapewniać, że system będzie skonteneryzowany, co będzie umożliwiać uruchomienie i konfigurację na różnych platformach.
- Aplikacja powinna zapewniać, że interfejs będzie intuicyjny, umożliwiający użytkownikom bezproblemowe wykonywanie operacji.

### 3.3 Przypadki użycia

#### 3.3.1 Biznesowe przypadki użycia

##### **PB1 - Rozpatrzenie prośby o rejestrację**

Aktorzy - browar komercyjny/kontraktowy, administrator

*Scenariusz główny:*

1. Pracownik browaru podaje dane browaru i wysyła prośbę o rejestrację do administratora.
2. Administrator analizuje dane browaru.
3. Administrator zezwala browarowi na dostęp do systemu.

*Scenariusz alternatywny:*

- 1-2. Jak w scenariuszu głównym.
3. Administrator odrzuca prośbę o rejestrację.

##### **PB2 - Administrator sprawdza statystyki systemu i browarów**

Aktorzy: administrator

*Scenariusz:*

1. Administrator otwiera panel statystyk.
2. Administrator wybiera jaką statystykę (lista browarów, ilość wyprodukowanego piwa, anulowane rezerwacje itp.) go interesuje.
3. Administrator może przeglądać te dane.

### **PB3 - Rezerwacja infrastruktury browaru komercyjnego**

Aktorzy - browar kontraktowy, browar komercyjny

*Scenariusz główny:*

1. Pracownik browaru kontraktowego określa parametry urządzeń, które chce wynająć.
2. Pracownik browaru kontraktowego określa, kiedy chce wynająć urządzenia.
3. Pracownik browaru kontraktowego wybiera interesujące go urządzenia.
4. Pracownik browaru kontraktowego określa, ile piwa zamierza wyprodukować.
5. Pracownik browaru kontraktowego określa listę osób, które będą zajmowały się wytwarzaniem piwa.
6. Pracownik browaru kontraktowego wysyła prośbę o rezerwację wybranych urządzeń.
7. Pracownik browaru komercyjnego analizuje prośbę o rezerwację.
8. Pracownik browaru komercyjnego akceptuje prośbę o rezerwację.

*Scenariusz alternatywny:*

- 1-7. Jak w scenariuszu głównym.
8. Pracownik browaru komercyjnego odrzuca prośbę o rezerwację.

*Scenariusz alternatywny:*

- 1-5. Jak w scenariuszu głównym.
6. Pracownik browaru kontraktowego wysyła prośbę o rezerwację wybranych urządzeń oraz zablokowanie dostępu do danego obszaru browaru komercyjnego na czas wykonania.
- 7-8. Jak w scenariuszu głównym.

### **PB4 - Anulowanie rezerwacji przez browar kontraktowy/komercyjny**

Aktorzy: browar kontraktowy/komercyjny

*Scenariusz:*

1. Pracownik browaru kontraktowego/komercyjnego wyszukuje rezerwację w harmonogramie lub liście.
2. Pracownik browaru kontraktowego/komercyjnego otwiera detale rezerwacji.
3. Pracownik browaru kontraktowego/komercyjnego anuluje rezerwację.

### **PB5 - Zarządzanie infrastrukturą przez browar komercyjny**

Aktorzy: browar komercyjny

Scenariusz:

1. Pracownik browaru komercyjnego wybiera czy chce dodać lub zmodyfikować urządzenie z wyświetlonej listy dostępnych urządzeń.
2. Pracownik browaru komercyjnego dodaje / modyfikuje parametry / usuwa urządzenia.

### **PB6 - Browar kontraktowy dodaje recepturę do systemu**

Aktorzy: browar kontraktowy

Scenariusz:

1. Pracownik browaru kontraktowego podaje do jakiego typu piwa odnosi się receptura.
2. Pracownik browaru kontraktowego wpisuje treść receptury.

### **PB7 - Browar kontraktowy wpisuje przebieg wykonania piwa do systemu**

Aktorzy: browar kontraktowy

Scenariusz:

1. Pracownik browaru kontraktowego danego dnia jest w trakcie wytwarzania piwa w browarze komercyjnym zgodnie z recepturą.
2. Pracownik browaru kontraktowego po wykonaniu danego etapu zapisuje czas i dokładny sposób jego przebiegu.
3. Po wykonaniu ostatniego etapu pracownik zapisuje, czy piwo zostało wytworzone z sukcesem.

**PB8 - Pracownik browaru komercyjnego sprawdza jacy pracownicy browaru kontraktowego mają wstęp na teren browaru komercyjnego**

Aktorzy: browar komercyjny

Scenariusz:

1. Pracownik browaru komercyjnego wprowadza datę.
2. Pracownik browaru komercyjnego odczytuje listę osób, na której podstawie stwierdza kto ma wstęp na teren browaru komercyjnego.

**PB9 - Pracownik browaru komercyjnego sprawdza harmonogram rezerwacji urządzeń**

Aktorzy: browar komercyjny

Scenariusz:

1. Pracownik browaru komercyjnego wybiera urządzenie z listy.
2. Pracownik browaru komercyjnego widzi przedziały dat, kiedy urządzenie jest zarezerwowane, przez jaki browar kontraktowy, jaki rodzaj piwa będzie w niej produkowany.

**PB10 - Pracownik browaru kontraktowego sprawdza ofertę infrastruktury browarów komercyjnych**

Aktorzy: browar kontraktowy

Scenariusz:

1. Pracownik browaru kontraktowego określa parametry urządzeń, które chce wynająć.
2. Pracownik browaru kontraktowego określa, kiedy chce wynająć urządzenia.
3. Pracownik browaru kontraktowego wybiera interesujące go urządzenia.
4. Pracownik browaru kontraktowego wychodzi z ekranu bez wysyłania prośby o rezerwację.

**PB11 - Pracownik browaru kontraktowego/komercyjnego sprawdza swoje aktywne rezerwacje**

Aktorzy: browar kontraktowy/komercyjny

Scenariusz:

1. Pracownik browaru kontraktowego/komercyjnego przechodzi do panelu ze swoimi aktywnymi rezerwacjami.
2. Pracownik browaru przegląda szczegóły wybranej aktywnej rezerwacji.

## **PB12 - Przeglądanie konta**

Aktorzy: browar komercyjny/browar kontraktowy/administrator

Scenariusz:

1. Użytkownik przechodzi do głównego pulpitu.
2. Użytkownik wybiera opcję, którą chce przeglądać.

### **3.3.2 Systemowe przypadki użycia**

#### **FU1 - Wyszukiwanie browarów oferujących dane urządzenia**

Wspiera procedurę PB3 - Rezerwacja infrastruktury browaru komercyjnego.

Aktorzy: browar kontraktowy

Scenariusz:

1. Pracownik browaru kontraktowego określa parametry urządzeń z poszczególnych etapów produkcji piwa, są to:
  - a. Dla etapu I
    - i. Wybicie
    - ii. Zakres pH wody w zakładzie
    - iii. Okres czasowy wypożyczenia
  - b. Dla etapu II
    - i. Wybicie
    - ii. Zakres temperatury
    - iii. Potrzeba użycia dwóch fermentorów
    - iv. Okres czasowy wypożyczenia
  - c. Dla etapu III
    - i. Wybicie
    - ii. Zakres temperatury
    - iii. Rodzaje rozlewu
    - iv. Okres czasowy wypożyczenia

Oprócz parametrów urządzeń, browar kontraktowy może określić przedział współczynnika anulowanych rezerwacji i współczynnika nieudanych procesów wytwarzania piwa w danym browarze komercyjnym.

2. System zwraca listę browarów, które oferują urządzenia zgodne z parametrami podanymi wcześniej.



## **FU2 - Złożenie prośby o rejestrację przez browar**

Wspiera procedurę PB1 - Rozpatrzenie prośby o rejestrację.

Aktorzy: browar komercyjny/kontraktowy

*Scenariusz:*

1. System wyświetla okno rejestracji.
2. Pracownik browaru wprowadza dane firmy (nazwę firmy) i dane konta (adres email, hasło).
3. System zapisuje prośbę o rejestrację.
4. System zamyka okno rejestracji.

## **FU3 - Rozpatrzenie prośby o rejestrację**

Wspiera procedurę PB1 - Rozpatrzenie prośby o rejestrację.

Aktorzy: administrator

*Scenariusz główny:*

1. System wyświetla okno aktywnych próśb o rejestrację.
2. Administrator wybiera jedną z próśb.
3. Administrator analizuje dane browaru.
4. Administrator akceptuje prośbę o zarejestrowanie.
5. System daje danemu browarowi dostęp do korzystania z aplikacji.
6. Powrót do kroku 1.

*Scenariusz alternatywny - odrzucenie prośby:*

- 1-3. Jak w scenariuszu głównym.
4. Administrator odrzuca prośbę o zarejestrowanie.
5. System nie daje danemu browarowi dostępu do korzystania z aplikacji.
6. Powrót do kroku 1 scenariusza głównego.

#### **FU4 - Logowanie browaru komercyjnego/kontraktowego**

Wspiera procedury PB1, PB3-PB10.

Aktorzy: browar komercyjny/kontraktowy

*Scenariusz główny:*

1. System wyświetla ekran logowania.
2. Pracownik browaru wpisuje email i hasło.
3. System sprawdza podane dane.
4. System loguje użytkownika.

*Scenariusz alternatywny - brak zgodności maila/hasła:*

- 1-3 Jak w scenariuszu głównym.
4. System wyświetla komunikat o podaniu niepoprawnych danych.
5. Powrót do punktu 1 scenariusza głównego.

*Scenariusz alternatywny - prośba o rejestrację browaru nie została jeszcze rozpatrzona przez administratora:*

- 1-3 Jak w scenariuszu głównym.
4. System wyświetla komunikat o braku weryfikacji prośby o rejestrację.
5. Powrót do punktu 1 scenariusza głównego.

*Scenariusz alternatywny - prośba o rejestrację browaru została odrzucona przez administratora:*

- 1-3 Jak w scenariuszu głównym.
4. System wyświetla komunikat o odrzuceniu prośby o rejestrację.
5. Powrót do punktu 1 scenariusza głównego.

## **FU5 - Rezerwacja zestawu urządzeń**

Wspiera procedurę PB3 - Rezerwacja infrastruktury browaru komercyjnego.  
Funkcja specjalizująca FU1. Korzysta z FU4.

### *Scenariusz główny:*

1. Pracownik browaru kontraktowego wybiera browar komercyjny od którego chce wypożyczyć sprzęt.
2. System pokazuje panel wyboru sprzętu. Sprzęt jest podzielony na 3 etapy:
  - a. Etap I - pracownik browaru kontraktowego wybiera urządzenie i określa czas na jaki chce je wypożyczyć. Po wybraniu urządzenia wyświetlają się jego szczegóły. W reszcie etapów przestają się wyświetlać opcje niekompatybilne z wybranym urządzeniem.
  - b. Etap II - pracownik browaru kontraktowego wybiera kadź dla fermentacji, opcjonalnie drugą kadź do drugiej fazy fermentacji oraz czasy ich wypożyczenia. Po wybraniu kadzi wyświetlają się ich szczegóły. W reszcie etapów przestają się wyświetlać opcje niekompatybilne z wybranymi kadziami.
  - c. Etap III - pracownik browaru kontraktowego wybiera kadź do leżakowania i czas jej wypożyczenia oraz sposób rozlewu. Po wybraniu kadzi do leżakowania wyświetlają się jej szczegóły. W reszcie etapów przestają się wyświetlać opcje niekompatybilne z wybraną kadzią do leżakowania i sposobem rozlewu.
3. Pracownik browaru kontraktowego wybiera sprzęty z każdego etapu.
4. System weryfikuje poprawność żądania.
5. System przenosi do podsumowania, które wyświetla wybrane sprzęty, czas ich wypożyczenia oraz wyliczoną cenę.
6. Pracownik browaru kontraktowego wysyła prośbę o wypożyczenie.

### *Scenariusz alternatywny - żądanie niepoprawne:*

- 1-4. Tak samo jak w scenariuszu głównym.
5. System wyświetla komunikat z błędem.
6. Powrót do punktu 3 scenariusza głównego.

## **FU6 - Anulowanie rezerwacji przez browar kontraktowy/komercyjny**

Wspiera procedurę PB4

Rozszerza funkcję FU18 po kroku 3.

Aktorzy: browar kontraktowy/komercyjny

*Scenariusz główny:*

1. System wyświetla okno aktywnych rezerwacji.
2. Pracownik browaru kontraktowego/komercyjnego wprowadza dane, na podstawie których chce znaleźć rezerwację.
3. System wyświetla listę rezerwacji spełniających dane kryteria.
4. Pracownik browaru kontraktowego/komercyjnego wybiera rezerwację.
5. System wyświetla detale rezerwacji.
6. Pracownik browaru kontraktowego/komercyjnego wybiera opcję anulowania rezerwacji.
7. System anuluje rezerwację.

*Scenariusz alternatywny - wybranie rezerwacji bez wcześniejszego wyszukiwania:*

1. Jak w scenariuszu głównym.
2. Powrót do kroku 4 scenariusza głównego.

## **FU7 - Logowanie administratora**

Wspiera procedury PB1, PB2.

Aktorzy: administrator

*Scenariusz główny:*

5. System wyświetla ekran logowania.
6. Administrator wpisuje email i hasło.
7. System sprawdza podane dane.
8. System loguje użytkownika.

*Scenariusz alternatywny - brak zgodności maila/hasła:*

- 1-3 Jak w scenariuszu głównym.
4. System wyświetla komunikat o podaniu niepoprawnych danych.
5. Powrót do punktu 1 scenariusza głównego.

## **FU8 - Dodanie nowego urządzenia przez browar komercyjny**

Wspiera procedurę PB5 - Zarządzanie infrastrukturą przez browar komercyjny.

Rozszerza funkcję FU11 po kroku 2 (w punkcie wybór operacji).

Aktorzy: browar komercyjny

*Scenariusz:*

1. System otwiera okno dodawania urządzeń.
2. Pracownik browaru komercyjnego wybiera kategorię urządzenia jakie chce dodać.
3. System wyświetla parametry możliwe do określenia dla tej kategorii urządzenia.
4. Pracownik browaru komercyjnego podaje parametry urządzenia.
5. Pracownik browaru komercyjnego podaje cenę za wypożyczenie urządzenia na jeden dzień.
6. Pracownik browaru komercyjnego podaje, ile tak skonfigurowanych urządzeń chce dodać.
7. System dodaje urządzenia do infrastruktury browaru i wyświetla komunikat potwierdzający.
8. Powrót do kroku 1.

## **FU9 - Dodanie receptury**

Wspiera procedurę PB6.

Aktorzy: browar kontraktowy

*Scenariusz:*

1. System otwiera okno dodania receptury.
2. Pracownik browaru kontraktowego podaje do jakiego typu piwa odnosi się receptura.
3. Pracownik browaru kontraktowego wpisuje treść receptury.
4. System dodaje recepturę do receptur browaru.

## **FU10 - Aktualizacja przebiegu wykonania piwa**

Wspiera procedurę PB7

Aktorzy: browar kontraktowy

*Scenariusz główny:*

1. System otwiera okno aktywnych procesów wytwarzania piwa.
2. Pracownik browaru kontraktowego wybiera który z nich chce uzupełnić.
3. System otwiera okno detali danego procesu wytwarzania piwa.
4. Pracownik browaru kontraktowego zapisuje czas wykonaniu danego etapu.
5. Pracownik browaru kontraktowego zapisuje dokładny sposób jego przebiegu etapu.
6. System aktualizuje dane danego przebiegu wykonania piwa.

*Scenariusz alternatywny - został wykonany ostatni etap wykonania piwa:*

- 1-5. Jak w scenariuszu głównym.
6. Pracownik browaru kontraktowego zapisuje, czy piwo zostało wytworzone z sukcesem.
7. Powrót do kroku 6 w scenariuszu głównym.

## **FU11 - Zarządzanie urządzeniami**

Wspiera procedurę PB5.

Rozszerza funkcję FU17 po kroku 2 (w punkcie wybór operacji).

Aktorzy: browar komercyjny

*Scenariusz główny:*

1. Pracownik wybiera okno zarządzania urządzeniami w jego browarze.
2. System otwiera okno zarządzania urządzeniami i wyświetla listę dostępnych urządzeń [wybór operacji].

## **FU12 - Usunięcie urządzenia przez browar komercyjny**

Wspiera procedurę PB5.

Rozszerza funkcję FU11 po kroku 2 (w punkcie wybór operacji).

Aktorzy: browar komercyjny

*Scenariusz główny:*

1. Pracownik wybiera urządzenie z listy.
2. System wyświetla okno ze szczegółami urządzenia.
3. Pracownik wybiera przycisk usuń.
4. System usuwa urządzenie z infrastruktury browaru komercyjnego i wyświetla okno potwierdzające wykonanie operacji.

*Scenariusz alternatywny - urządzenie jest zarezerwowane przez browar kontraktowy:*

- 1 - 3. Jak w scenariuszu głównym.
4. System wyświetla okno informujące o konieczności anulowania rezerwacji przed usunięciem danego urządzenia.

## **FU13 - Modyfikacja danych urządzenia przez browar komercyjny**

Wspiera procedurę PB5.

Rozszerza funkcję FU11 po kroku 2 (w punkcie wybór operacji).

Aktorzy: browar komercyjny

*Scenariusz główny:*

1. Pracownik wybiera urządzenie z listy.
2. System wyświetla okno ze szczegółami urządzenia.
3. Pracownik wybiera przycisk modyfikuj.
4. System wyświetla formularz z aktualnymi parametrami urządzenia, które można modyfikować.

*Scenariusz alternatywny - urządzenie jest zarezerwowane:*

- 1 - 3. Jak w scenariuszu głównym.
4. System wyświetla ostrzeżenie informację o aktualnej rezerwacji.
5. System wyświetla formularz z aktualnymi parametrami urządzenia, które można modyfikować.

#### **FU14 - Wyłączenie urządzenia z użytku**

Wspiera procedurę PB5.

Rozszerza funkcję FU11 po kroku 2 (w punkcie wybór operacji).

Aktorzy: browar komercyjny

*Scenariusz główny:*

1. Pracownik wybiera urządzenie z listy.
2. System wyświetla okno ze szczegółami urządzenia.
3. Pracownik wybiera przycisk wyłącz z użytku.
4. System wyświetla okno z możliwością wybrania daty, od której urządzenie ma być wyłączone opcjonalnie datę zakończenia wyłączenia.
5. System dodaje wyłączenie z użycia do harmonogramu urządzenia.

*Scenariusz alternatywny - urządzenie jest zarezerwowane przez browar kontraktowy:*

- 1 - 4. Jak w scenariuszu głównym.
5. System wyświetla okno informujące o konieczności anulowania rezerwacji przed wyłączeniem z użytku danego urządzenia.

#### **FU15 - Sprawdzenie jacy pracownicy są uprawnieni do wstępu do browaru**

Wspiera procedurę PB8.

Rozszerza funkcję FU17 po kroku 2 (w punkcie wybór operacji).

Aktorzy: browar komercyjny

*Scenariusz główny:*

1. System wyświetla odpowiednie okno.
2. Pracownik browaru komercyjnego wprowadza datę.
3. System wyświetla listę osób, które mają wstęp na teren browaru komercyjnego w danym dniu.
4. Pracownik browaru komercyjnego odczytuje listę pracowników.

*Scenariusz alternatywny - brak pracowników mających wstęp w danym dniu:*

- 1-2. Jak w scenariuszu głównym.
3. System informuje, że nie ma osób, które mają wstęp na teren browaru komercyjnego w danym dniu.



### **FU16 - Sprawdzenie harmonogramu rezerwacji urządzenia**

Wspiera procedurę PB9.

Rozszerza funkcję FU11 po kroku 2 (w punkcie wybór operacji).

Aktorzy: browar komercyjny

*Scenariusz:*

1. Pracownik wybiera urządzenie z listy.
2. System wyświetla okno ze szczegółami urządzenia.
3. Pracownik wybiera przycisk harmonogram.
4. System wyświetla listę z terminami, w których urządzenie jest zarezerwowane, wraz z odnośnikiem do tej rezerwacji.

### **FU17 - Przeglądanie konta**

Wspiera procedurę PB12.

Aktorzy: administrator/browar komercyjny/kontraktowy

*Scenariusz:*

1. Użytkownik loguje się za pomocą funkcji FU4 lub FU7.
2. System wyświetla główne okno z różnymi opcjami zależnymi od użytkownika [wybór operacji].

### **FU18 - Pracownik browaru sprawdza swoje aktywne rezerwacje**

Wspiera procedurę PB11.

Aktorzy: browar kontraktowy/browar komercyjny

*Scenariusz:*

1. Pracownik browaru przechodzi do panelu ze swoimi aktywnymi rezerwacjami.
2. System wyświetla aktywne rezerwacje.
3. Pracownik browaru przegląda szczegóły wybranej aktywnej rezerwacji.

#### **FU19 - Administrator sprawdza statystyki systemu i browarów**

Wspiera procedurę PB2.

Aktorzy: administrator

*Scenariusz:*

1. Administrator wybiera opcję wyświetlania statystyk.
2. System otwiera okno statystyk.
3. Administrator może filtrować po rodzaju browaru.
4. Administrator może sortować po ilości wyprodukowanego piwa, procentowym stosunku rezerwacji anulowanych przez browar do wszystkich rezerwacji browaru, procentowym stosunku nieudanych procesów wytwarzania piwa do wszystkich procesów wytwarzania piwa.
5. System wyświetla listę browarów wedle wybranych filtrów oraz sumarycznie.

#### **FU20 - Administrator sprawdza statystyki pojedynczego browaru**

Wspiera procedurę PB2.

Rozszerza funkcję FU20 po kroku 4.

*Scenariusz:*

1. Administrator wybiera pojedynczy browar z listy.
2. System wyświetla okno ze statystykami tego browaru

## FU21 - Pracownik browaru komercyjnego sprawdza prośby o rezerwację

Wspiera procedury PB3, PB11.

Aktorzy: browar komercyjny

*Scenariusz główny:*

1. Pracownik browaru przechodzi do panelu z prośbami o rezerwację.
2. System wyświetla listę prośb o rezerwację.
3. Pracownik browaru przegląda szczegóły wybranej rezerwacji. W szczegółach wyświetlają się dane o urządzeniach i ich czasie wykorzystania, a także współczynnik anulowania rezerwacji browaru kontraktowego od którego pochodzi prośba o rezerwację.
4. Pracownik browaru akceptuje wybraną prośbę o rezerwację.

*Scenariusz alternatywny:*

- 1-3. Jak w scenariuszu głównym.
4. Pracownik browaru odrzuca wybraną prośbę o rezerwację.

### 3.3.3 Diagramy przypadków użycia

Diagramy ilustrują relacje między systemowymi przypadkami użycia dla różnych użytkowników. Przypadki są przedstawione zgodnie z notacją UML.

Diagram przypadków użycia dla pracownika browaru kontraktowego:

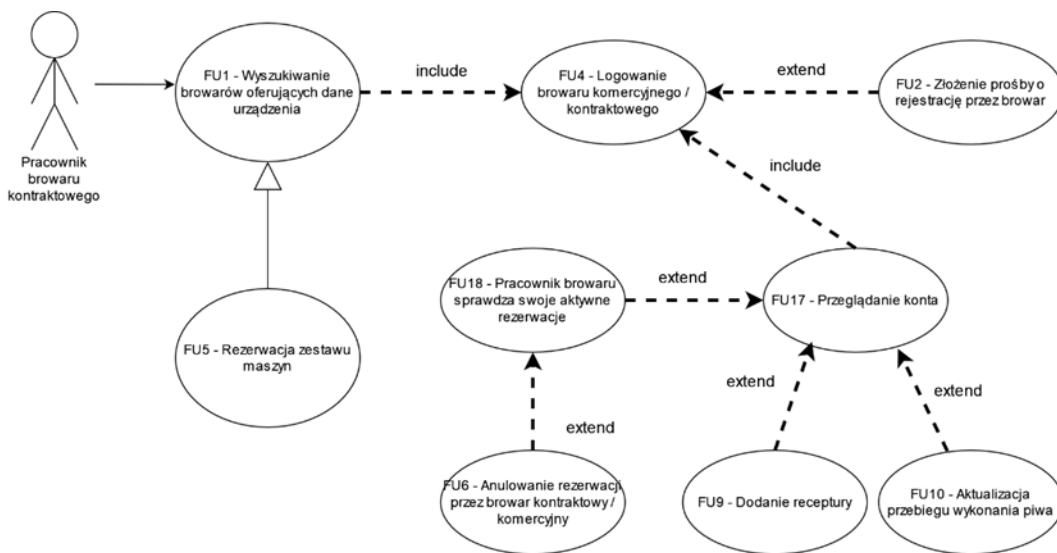


Diagram przypadków użycia dla pracownika browaru komercyjnego:

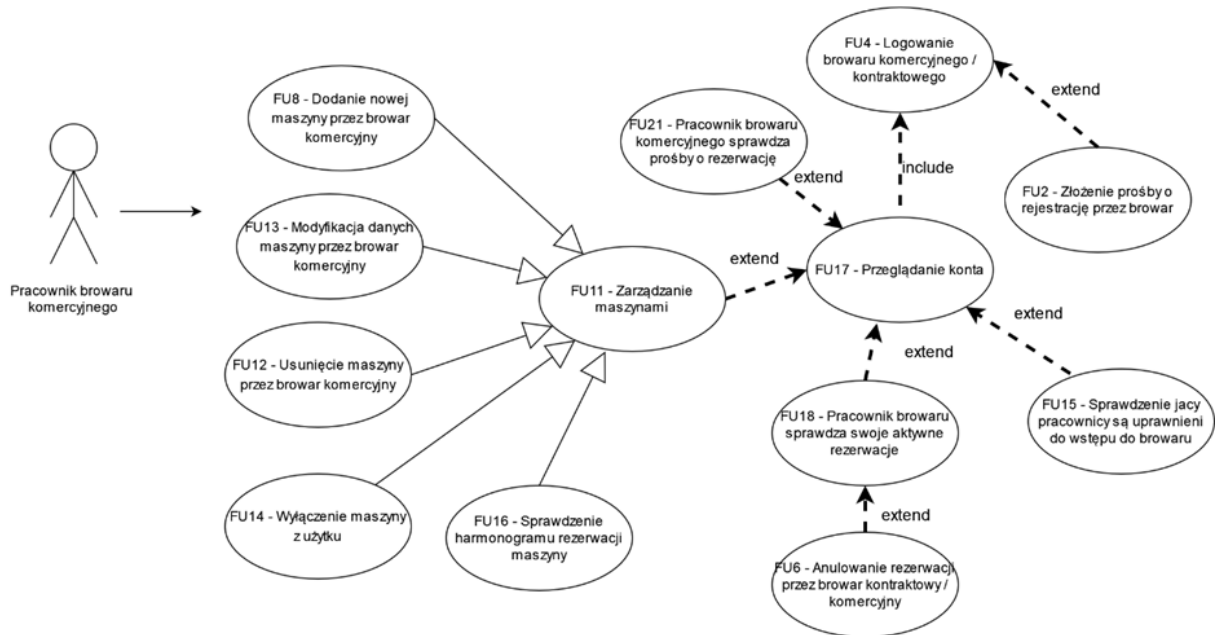
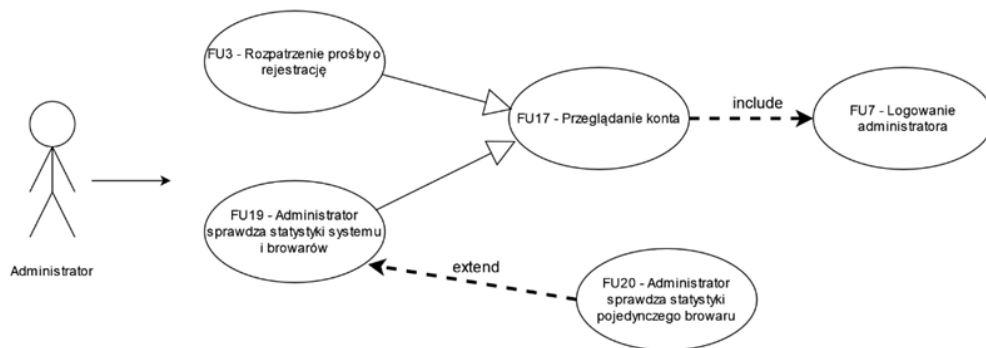


Diagram przypadków użycia dla administratora:



### 3.4 Potwierdzenie zgodności wymagań

Dzień dobry Państwu,

Dziękuję za wymagania, są zapisane bardzo dobrze. Mam tylko uwagę:

- wymaganie: „Aplikacja powinna pozwalać browarom kontraktowym na określenie, ile piwa zamierzają wyprodukować.”

-> „Aplikacja powinna pozwalać browarom kontraktowym na planowanie produkcji piwa”

Kind regards,

Piotr Pałka



Cabaj Krzysztof Yesterday 16:29

Zespół 12

Dokumentacja mi się podoba - kilka drobiazgów edycyjnych zaznaczyłem w tekście.

Pozdrawiam,  
Krzysztof Cabaj

## 4 Definicja architektury

### 4.1 Plan struktury systemu

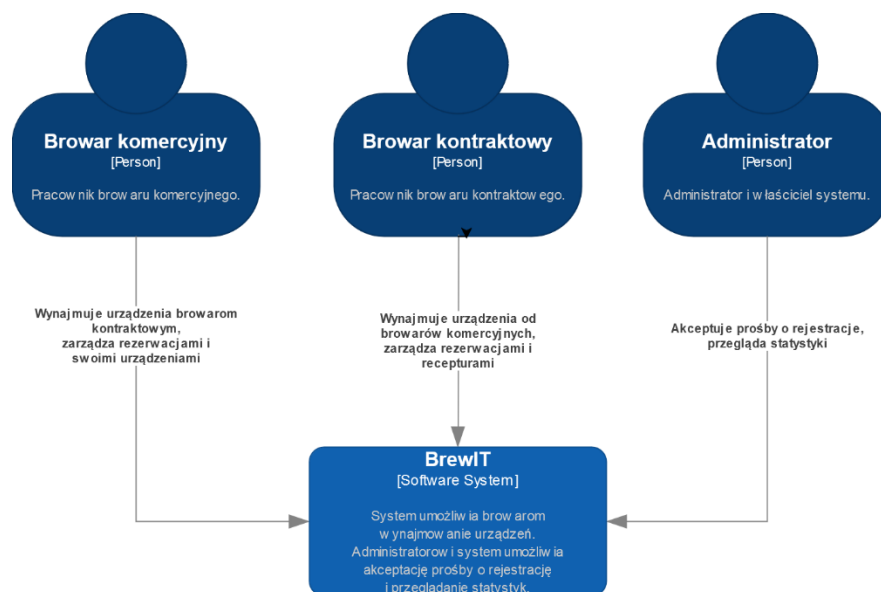
Plan struktury systemu zamodelowaliśmy przy użyciu modelu C4. Model ten ma hierarchiczną strukturę z czterema warstwami:

- diagram systemów,
- diagram kontenerów,
- diagram komponentów,
- diagram kodu.

Nasz plan struktury systemu nie obejmuje diagramu kodu zgodnie z zaleceniami eksperta. Jednakże, aby zamodelować sposób w jaki system będzie wdrożony, stworzyliśmy jeszcze diagram wdrożenia, który opisuje zależności pomiędzy perspektywą logiczną i sprzętową.

#### 4.1.1 Diagram systemów

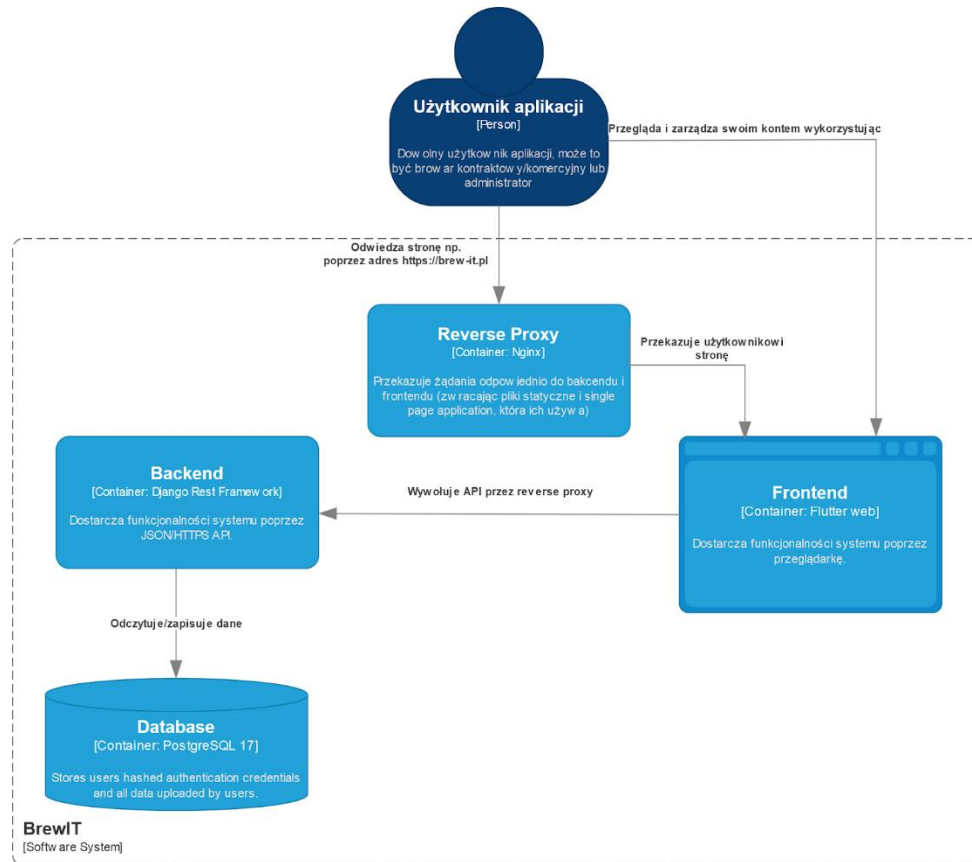
Tak prezentuje się diagram systemów.



Na diagramie można zauważyć 3 aktorów, użytkowników naszego systemu, którzy wykorzystują system w różny sposób. System BrewIT jest jedyny na diagramie, nie wykorzystujemy żadnego innego systemu zewnętrznego w naszej architekturze.

#### 4.1.2 Diagram kontenerów

Tak prezentuje się diagram kontenerów.



Użytkownik będzie wysyłać zapytania do serwera reverse proxy, którym będzie nginx. Nginx będzie zwracać stronę typu one page wraz z plikami statycznymi, które są przez nią używane. Nginx będzie także przekierowywać zapytania API przesyłane przez stronę do backendu.

Frontend to strona typu one page, zbudowana za pomocą frameworka flutter web. Będzie ona umożliwiała użytkownikom na interakcję z systemem BrewIT.

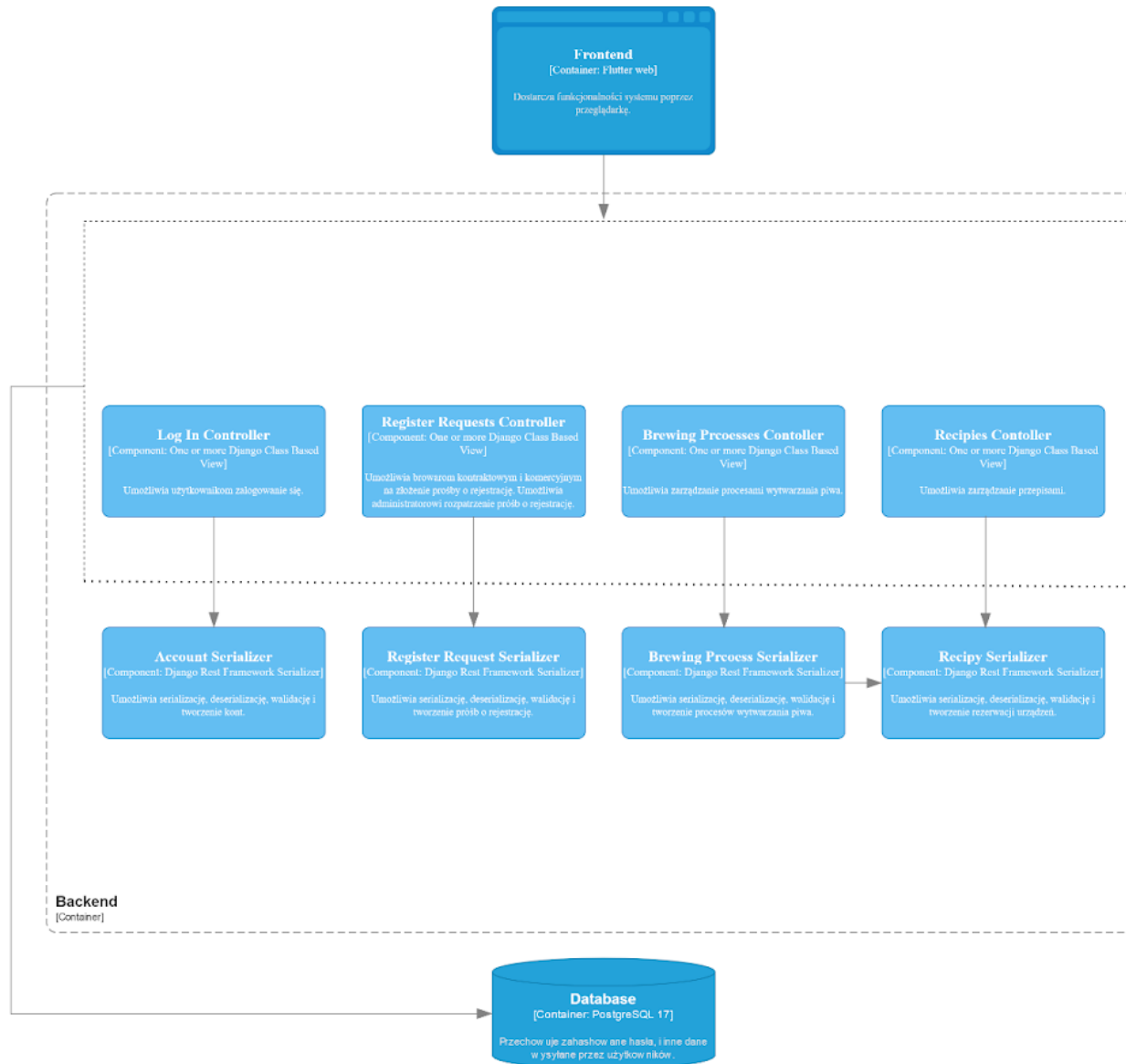
Backend to aplikacja napisana w języku Python z wykorzystaniem frameworka Django wraz z Django Rest. Aplikacja wystawia API, które będzie konsumowane przez frontend.

Baza danych będzie korzystać z PostgreSQL, backend będzie komunikować się z bazą danych za pomocą adaptera psycopg2.

### 4.1.3 Diagram komponentów

#### 4.1.3.1 Backend

Pierwsza część diagramu komponentów backendu.



Drużę część diagramu komponentów backendu.

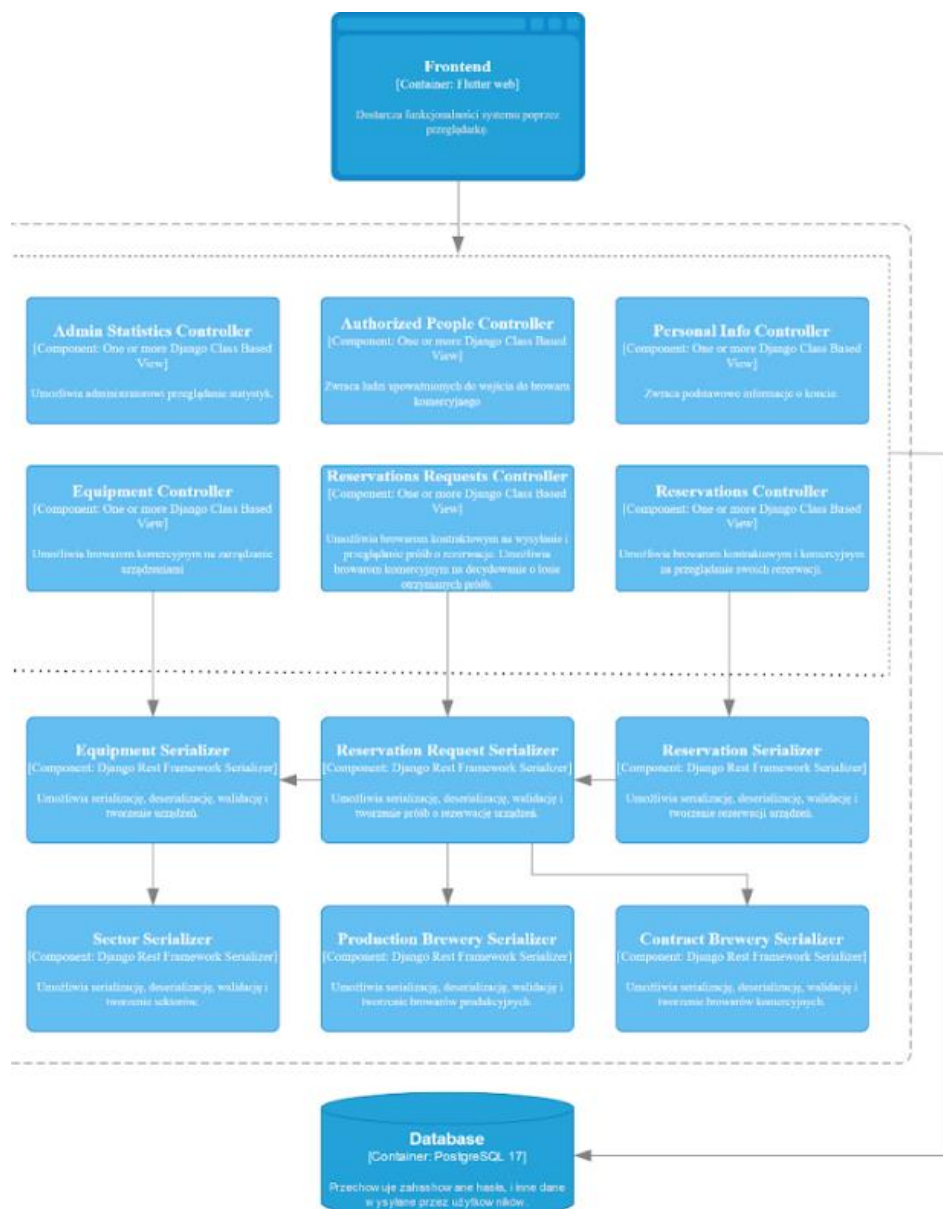


Diagram podzielono, gdyż był zbyt duży, by umieścić go w całości w pliku pdf. Przedstawia on komponenty backendu i sposób w jaki są powiązane z innymi kontenerami. Komponenty mają dwa typy - Controller i Serializer.

Controller jest to element, który nie występuje dosłownie w kodzie. Reprezentuje on kod, który odpowiada za daną funkcjonalność, w jego skład wchodzi jedna lub najczęściej więcej klas - wykorzystywanych w django class based views. Klasa taka implementuje pojedynczy API endpoint.

Serializer jest to klasa w Django Rest Framework, która implementuje serializację, deserializację oraz walidację danych, najczęściej zadeklarowanych modeli.



Jak można zauważyć na diagramie, wszystkie kontrolery odpowiadają na żądania API przesyłane przez frontend oraz używają bazy danych do zapisywania/odczytywania danych.

#### 4.1.3.2 Frontend

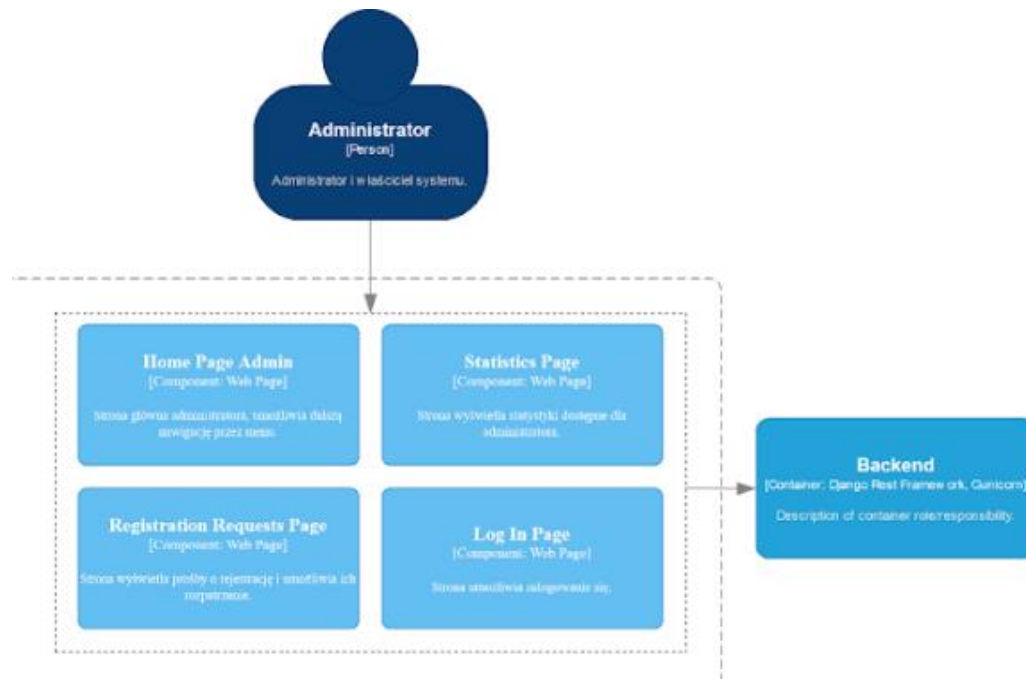
Pierwsza część diagramu komponentów frontendu.



Druga część diagramu komponentów frontendu.



Trzecia część diagramu komponentów frontendu.



Tak jak poprzednio podzielono diagram na części ze względu na jego duży rozmiar. Przedstawia on komponenty frontendu. Każdy komponent to strona, na którą może wejść użytkownik. W diagramie występuje trzech aktorów, każdy z nich używa innych stron do interakcji z systemem. Każdy z komponentów konsumuje API wystawione przez backend.

## 4.2 Szablony architektoniczne

W systemie wykorzystano szablon model-view-controller. W przypadku naszego systemu model to modele zadeklarowane w backendzie przy użyciu frameworku Django. Modele te reprezentują tabele w bazie danych, które zostały na ich podstawie stworzone.

View w naszym systemie to widoki, które dostarcza strona typu one page. Wyświetlają one dane przechowywane w bazie danych i umożliwiają konsumowanie API wystawione przez backend.

Controller w naszym systemie to class based views, używane w aplikacji Django. Przetwarzają one żądania API, wykonują operacje na danych i zwracają odpowiedź.

## 4.3 Wdrożenie

### Diagram wdrożenia

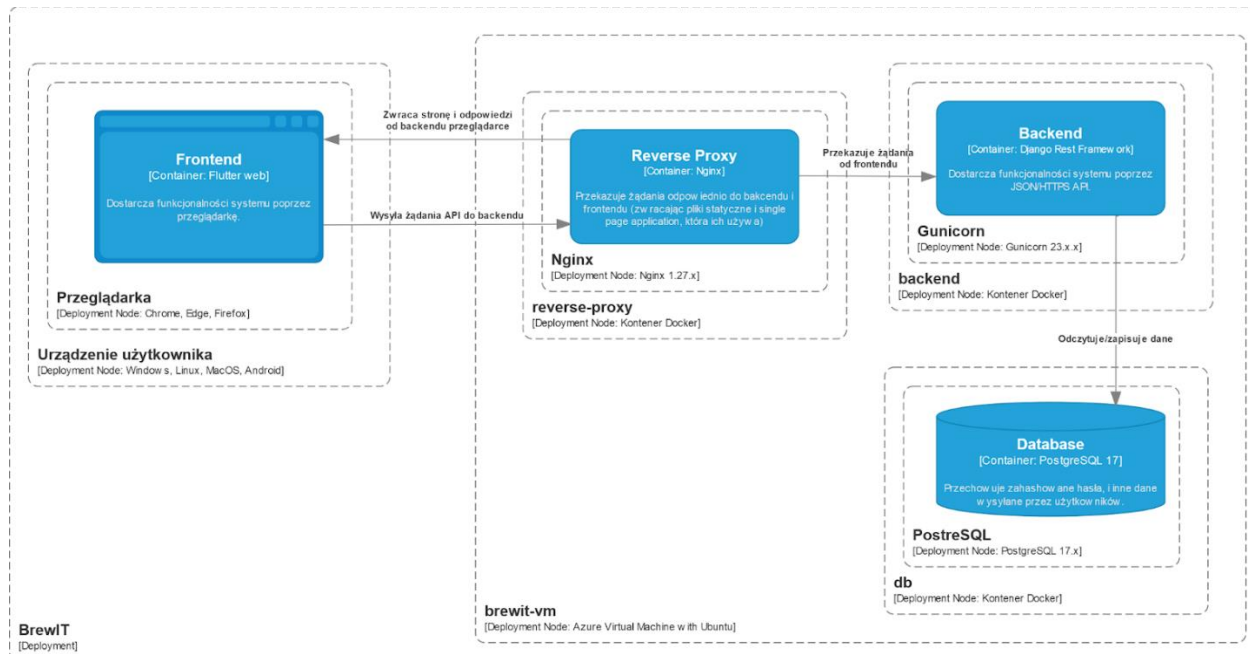
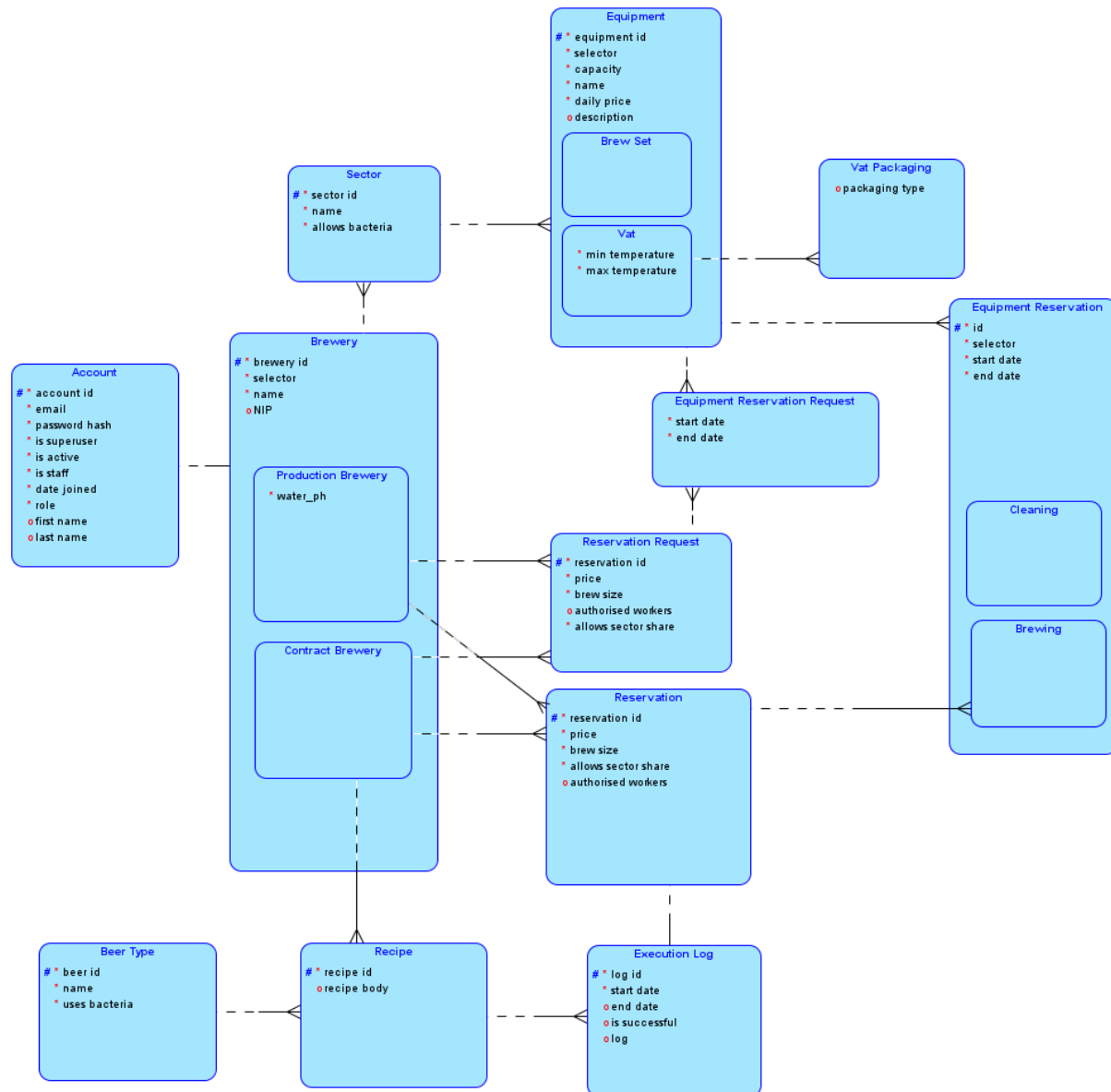


Diagram wdrożenia przedstawia jak komponenty logiczne będą rozmieszczone fizycznie. Wszystkie komponenty systemu będą znajdować się na maszynie wirtualnej azure z systemem Ubuntu. Reverse Proxy będzie używać nginx i będzie znajdować się w oddzielnym kontenerze dockerowym. Reverse Proxy będzie posiadało pliki statyczne oraz stronę typu one page, która będzie wysyłana do przeglądarki użytkownika. Backend będzie korzystał z serwera Gunicorn i będzie znajdować się w oddzielnym kontenerze dockerowym. Baza danych będzie korzystała z PostgreSQL i będzie znajdować się w oddzielnym kontenerze dockerowym.

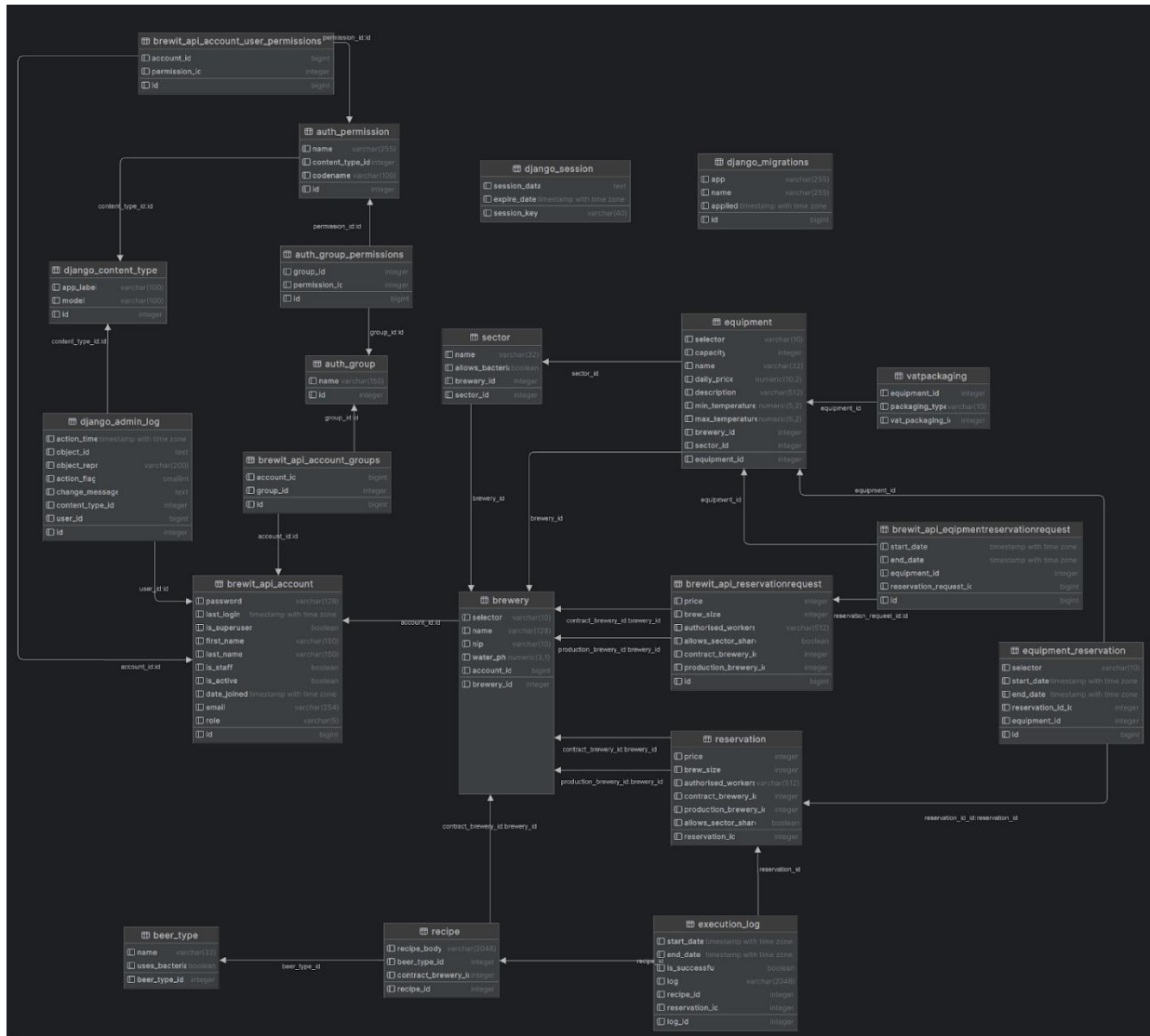
## 5 Dane trwałe

### 5.1 Model logiczny danych

#### Model ER



Model Relacyjny (Wygenerowany przy użyciu narzędzia DataGrip, zawiera również encje stworzone w Django, których nie ma w diagramie ER)



## Decyzje projektowe

Wybrana została generyczna implementacja podtypów, ponieważ ich tabele różnią się tylko jednym lub dwoma wierszami. Kolumny i relacje występujące dla tylko jednego z podtypów będą kontrolowane przy użyciu SQL constraints.

Encje:

Account - może być tylko jedno dla browaru.

Execution Log - został połączony z rezerwacją w celu zbierania informacji o powodzeniu wykonania rezerwacji w danym browarze.

Equipment - podtyp Brew Set nie różni się od nadtypu Equipment atrybutami i relacjami, ale został oddzielony ze względu na inną obsługę w aplikacji od podtypu Vat.

Equipment Reservation - podobnie jak w encji equipment, podtyp cleaning nie różni się od nadtypu, ale jest inaczej obsługiwany w aplikacji.

Reservation - posiada dwa klucze obce do encji brewing dla dwóch różnych relacji: z browarem produkcyjnym i kontraktowym.

## 5.2 Przetwarzanie i przechowywanie danych

Dane są przechowywane w relacyjnej bazie danych PostgreSQL. Została ona wybrana ze względu na wsparcie zaawansowanych funkcji, jej popularność na rynku baz relacyjnych, jej charakter open-source oraz wcześniejsze doświadczenie członków zespołu.

### Komunikacja z bazą danych

- Aplikacja komunikuje się z bazą danych za pomocą frameworka Django, który zapewnia warstwę ORM. ORM umożliwia manipulację danymi w bazie przy użyciu modeli obiektowych, co zwiększa czytelność i ułatwia zarządzanie kodem.

### Programowanie w bazie danych

- Logika aplikacji jest implementowana w warstwie aplikacyjnej, przy użyciu kodu napisanego w Pythonie i Django. Programowanie w bazie danych (funkcje, procedury składowane czy wyzwalacze) nie jest planowane, aby zapewnić większą czytelność i prostsze zarządzanie kodem.

### Indeksy

- Zostały dodane indeksy unikalne na wszystkich atrybutach unikalnych, czyli w przypadku naszej bazy danych na kluczach głównych i obcych.
- Indeksy na kolumnach, na których często będą wykonywane zapytania były rozważane, ale po wykonaniu testów podjęta została decyzja o nie korzystaniu z nich.

Wykonane zostały testy przy użyciu polecenia SQL EXPLAIN dla zapytań na tabeli equipment, na której najczęściej będzie stosowane wyszukiwanie z filtrowaniem. W tabeli zostało umieszczone przykładowe 1000 wierszy danych i dla każdego przypadku wykonane 5 identycznych zapytań. Przed każdym zapytaniem tabela była usuwana i dodawana ponownie, aby ograniczyć wpływ pamięci podręcznej.

	Z indeksem	Bez indeksu
średni czas przygotowania	0.182ms	0.142ms
średni czas wykonania	0.193ms	0.146ms

Zarówno czas przygotowania, jak i czas wykonania były niższe bez narzutu indeksów.

Repozytorium kodu: <https://github.com/angost/PZSP2-Zespol12-BrewIT>

Backend - język Python, framework Django.

Frontend - język Dart, framework Flutter.

Virtualenv do izolacji środowiska Pythonowego, w pliku requirements.txt znajdują się wymagane pakiety.

Środowisko uruchamiania - docker, utworzono plik *docker-compose.yml*, który definiuje kontenery dla backendu, bazy danych, frontendu i dodatkowych usług - nginx.

[illegible]



### 6.3 Statystyki

Liczba plików oraz linii została policzona za pomocą programu cloc.

Stan projektu na dzień 10/01/2025

21 testów jednostkowych

Komenda: `cloc . --exclude-dir=build,.dart_tool,.vscode,env`

Language	files	blank	comment	code
Python	30	468	131	2363
Dart	55	131	50	2349
YAML	3	12	22	64
JSON	1	0	0	35
HTML	1	7	12	19
Dockerfile	2	11	4	11
Markdown	1	2	0	3
SUM:	93	631	219	4844

## 7 Projekt standardu interfejsu użytkownika

Korzystając z narzędzia Figma stworzyliśmy Wireframe'y - wszystkie potrzebne w aplikacji ekrany, Wireflow'y - bazujące na zapisanych wcześniej przypadkach użycia oraz Mockup'y - dla 3 typów użytkowników.

Ekrany są spójne, stworzone na podstawie wymyślnego wcześniej Design Systemu. Wykorzystane są komponenty, warianty, interakcje, style tekstu i kolorów, siatki.

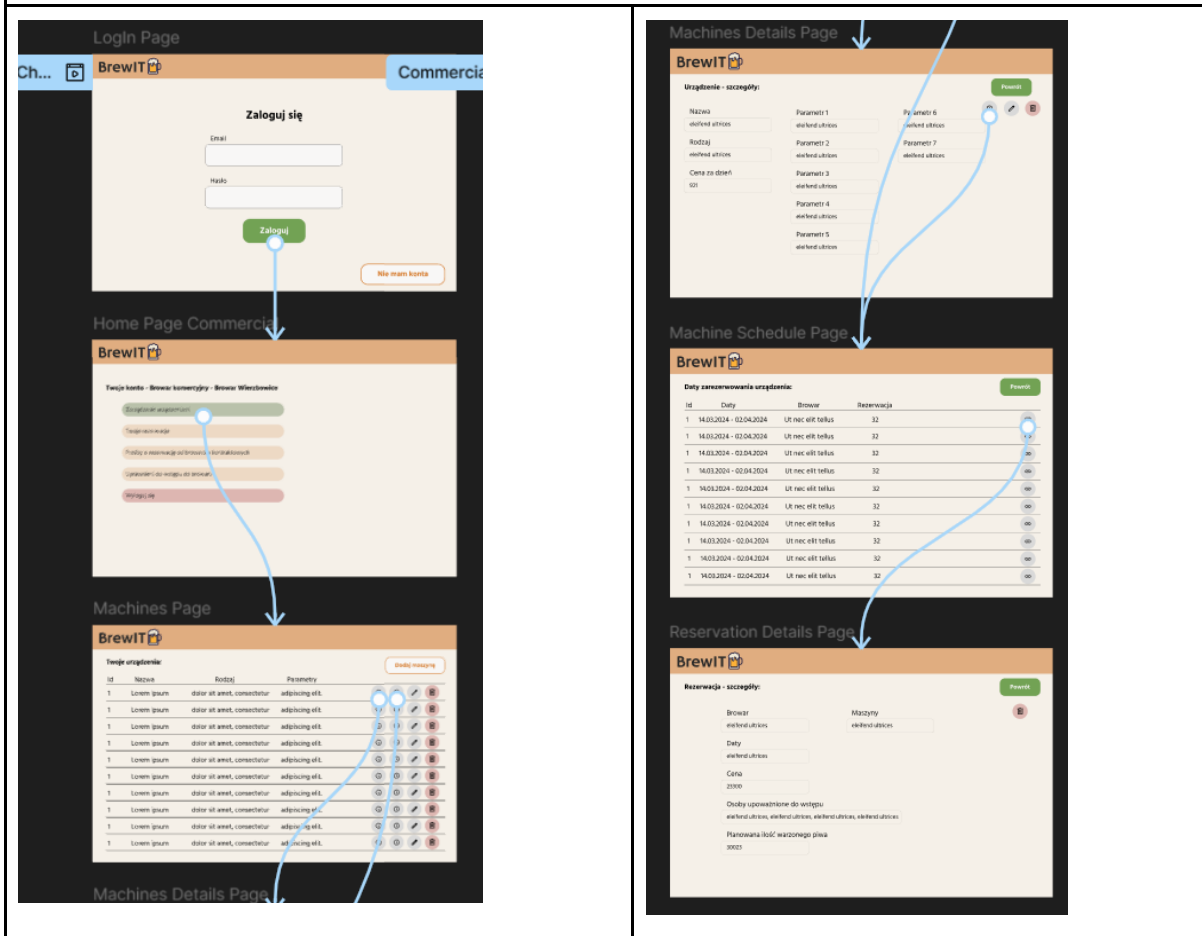
Link do projektu w Figma:

<https://www.figma.com/design/cHu79LPYPkiegyMd1VU8Oc/PZSP2-BrewIT-UI-UX?node-id=0-1&t=qHD3FLQqKMU2YGjF-1>

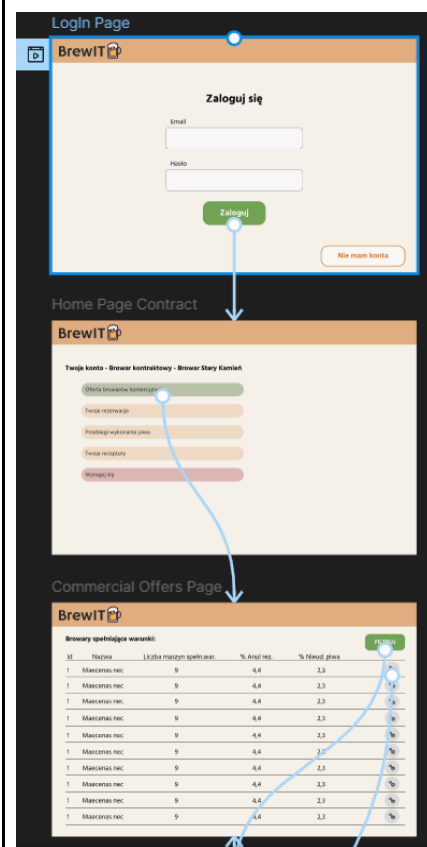
Przykładowe Wireflow'y występujące w Figma:

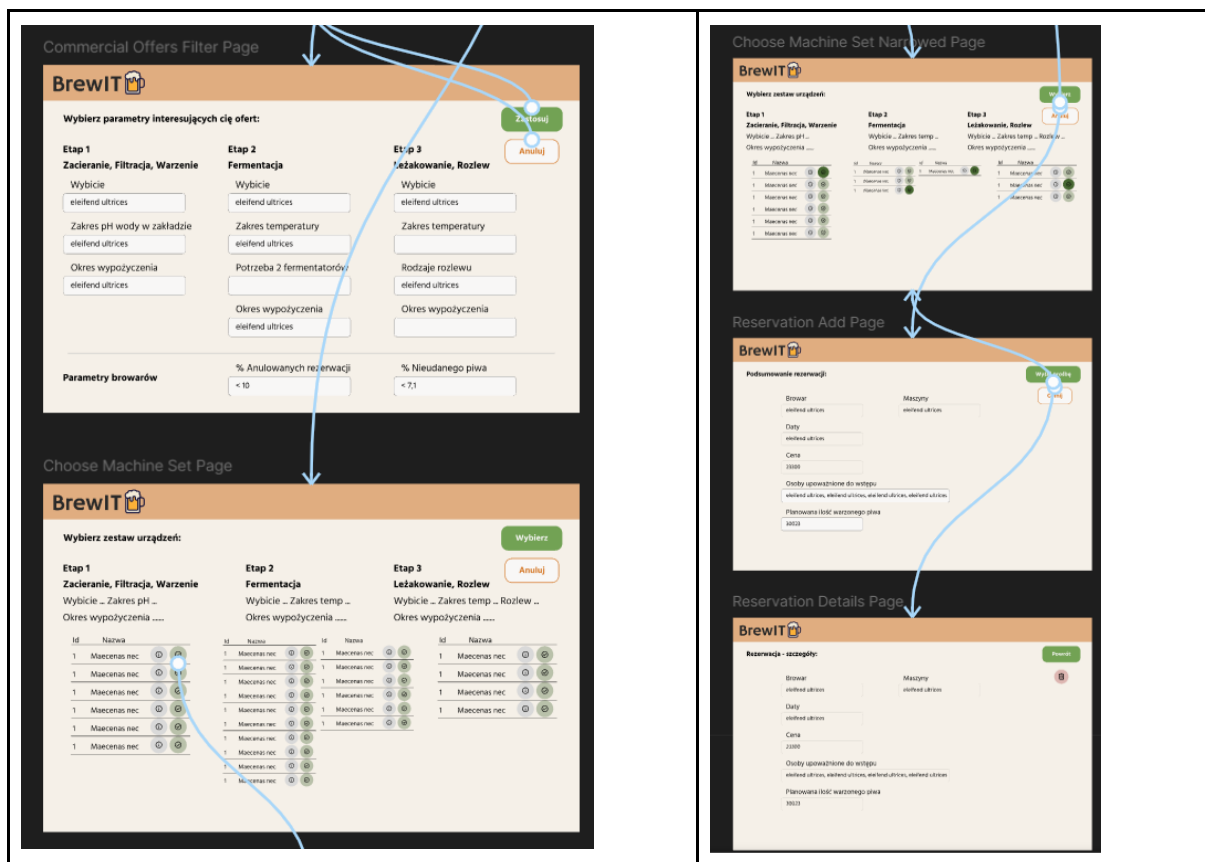


## FU16 - Sprawdzenie harmonogramu rezerwacji urządzenia



## 43





Stworzyliśmy Persony ilustrujące typowych użytkowników aplikacji.



### Marek - Menadżer browaru komercyjnego

- 40 lat
- Menadżer
- z Warszawy

#### Notka biograficzna

Marek ukończył Zarządzanie w Szkole Głównej Handlowej w Warszawie. W czasach studenckich zainteresował się piwem kraftowym. Zatrudnił się w pewnym browarze kontraktowym, a dzięki swojej ambicji wspiął się na menadżerskie stanowisko w firmie.

#### Osobowość

- Zorganizowany
- Towarzyski
- Inteligentny
- Pewny siebie

#### Zainteresowania

- Piwo kraftowe
- Nowe technologie

#### Motywacje

- Nie lubi tracić czasu, chce wykonywać swoją pracę jak najbardziej wydajnie
- Chce mieć więcej czasu do spędzania z rodziną
- Irytuje go konieczność poszukiwania klientów (browarów kontraktowych)

#### Cele

Dlaczego klient korzysta z systemu?

Usprawnienie procesu poszukiwania klientów

Usprawnienie procesu umawiania się z klientami na użyczenie usług browaru

#### Potrzeby

Czego klient oczekuje od systemu?

Możliwość umawiania rezerwacji bez konieczności kontaktu

Łatwe sprawdzanie przyszłych rezerwacji i ewentualne anulowanie ich

Automatyczne tworzenie harmonogramu urządzeń



## Piotr - Menadżer browaru kontraktowego

- 58 lat
- Menadżer
- z Bydgoszczy

### Notka biograficzna

Piotr od zawsze marzył o założeniu własnego browaru. Warzyć piwo nauczył się od dziadka. Do niedawna pracował jako mechanik, ale postanowił się przebranżwić. Nie ma ogromnych środków finansowych, ale mimo to chce rozwinąć swój biznes i rozprowadzić rodzinny przepis.

### Osobowość

- Z pasją
- Marzyciel
- Staromodny
- Nie jest za dobrze zapoznany z komputerem

### Zainteresowania

- Piwo kraftowe
- Nowe technologie

### Motywacje

- Nie wie gdzie szukać browarów, które mogłyby wykonać jego recepturę
- Zatrudnia kilku pracowników, ale nie stać go na budowę własnego, w pełni wyposażonego browaru

**Cele** Dlaczego klient korzysta z systemu?

Możliwość produkcji piwa bez posiadania własnych urządzeń

Znalezienie browarów, które mogą wykonać jego recepturę

**Potrzeby** Czego klient oczekuje od systemu?

Bazy ofert browarów u których może wykonać rezerwację

Dostosowanie browaru do jego potrzeb, w tym zapewnienie że jego piwo nie będzie miało kontaktu z piwami kwaśnymi

Kalendarza przyszłych i przeszłych rezerwacji

Klarownej wyceny kosztów rezerwacji



## Adam - Administrator (właściciel aplikacji)

- 32 lata
- Biznesmen
- z Wrocławia

### Notka biograficzna

Adam jest przedsiębiorczy, ma dobre i realne pomysły, założył już kilka startupów, które odniosły sukces na rynku. Ma wiedzę na temat piwa i chciałby ją wykorzystać do stworzenia systemu, na którym może zarobić.

### Osobowość

- Przedsiębiorczy
- Konkretny
- Introwertyk

### Zainteresowania

- Pasjonat technologii
- Networking
- Rozwój osobisty

### Motywacje

- Nie wie gdzie szukać browarów, które mogłyby wykonać jego recepturę
- Zatrudnia kilku pracowników, ale nie stać go na budowę własnego, w pełni wyposażonego browaru

**Cele** Dlaczego klient korzysta z systemu?

Zapewnienie że z systemu korzystać będą poważne firmy

Monitorowanie sukcesu aplikacji

**Potrzeby** Czego klient oczekuje od systemu?

Możliwość weryfikacji użytkowników

Sprawdzenie różnego rodzaju statystyk, takich jak liczba zarejestrowanych browarów komercyjnych i kontraktowych

Dobre spełnianie swoich podstawowych funkcji, aby odniósł sukces komercyjny

Przygotowaliśmy również przykładowe historyjki użytkowników, które są prostszym opisem przypadków użycia przygotowanych wcześniej.

Przykładowe historyjki użytkowników:

<b>Administrator</b>	
"Chcę sprawdzić statystyki aplikacji. Potrzebuję wiedzieć ilu użytkowników (browarów komercyjnych i kontraktowych) z niej korzysta i ile piwa sumarycznie jest produkowane. Chcę wiedzieć jaki jest średni % nieudanych produkcji piwa i średni % odwoływanych rezerwacji."	"PB2 - Administrator sprawdza statystyki systemu i browarów"
"Chcę przejrzeć aktywne prośby o zarejestrowanie w systemie, zobaczyć informacje które użytkownik wprowadził podczas rejestracji. Na tej podstawie chcę móc zaakceptować lub odrzucić tę prośbę."	"PB1 - Rozpatrzenie prośby o rejestrację"
<b>Browar komercyjny</b>	
"Chcę dodać do systemu nową maszynę. Potrzebuję móc określić jej typ, parametry i cenę za wypożyczenie."	"PB5 - Zarządzanie infrastrukturą przez browar komercyjny"
"Chcę sprawdzić jakie osoby są upoważnione do wejścia na teren browaru w danym dniu."	"PB8 - Pracownik browaru komercyjnego sprawdza jacy pracownicy browaru kontraktowego mają wstęp na teren browaru komercyjnego"
<b>Browar kontraktowy</b>	
"Chcę zarejestrować się w aplikacji."	"PB1 - Rozpatrzenie prośby o rejestrację"
"Chcę przejrzeć ofertę browarów komercyjnych, upewniając się, że jest ona dostosowana do moich potrzeb. Potrzebuję m.in. maszyn, które mają konkretny zakres temperatur, lub konkretny sposób rozlewu. Chcę też upewnić się, że browar, u którego zawieram rezerwację jest godny zaufania - ma odpowiedni współczynnik udanych procesów wykonania piwa i odpowiednio niski współczynnik anulowanych rezerwacji."	"PB10 - Pracownik browaru kontraktowego sprawdza ofertę infrastruktury browarów komercyjnych"



## 8 Specyfikacja testów

### 8.1 Rodzaje testów

Testy jednostkowe zostały przeprowadzone wbudowanym w Django frameworkiem, który zbudowany jest na module unittest.

Testy jednostkowe - automatycznie sprawdzają poprawność tworzonych modeli danych.

Testy integracyjne - przeprowadzone ręcznie. Planowane napisanie testów automatycznych dla endpointów api.

Testy systemowe - przeprowadzone ręcznie. Planowane napisanie testów automatycznych dla najważniejszych funkcji systemu.

### 8.2 Miara jakości testów:

Pomiary uzyskane za pomocą narzędzia *coverage*. Pomiary aktualne na dzień 10/01/2025.

Name	Stmts	Miss	Cover
-----	-----	-----	-----
brewit_api\admin.py	76	0	100%
brewit_api\apps.py	4	0	100%
brewit_api\auth_class.py	4	1	75%
brewit_api\filters.py	19	0	100%
brewit_api\managers.py	20	3	85%
brewit_api\models.py	133	0	100%
brewit_api\permissions.py	14	3	79%
brewit_api\serializers.py	243	119	51%
brewit_api\tests\test_models.py	176	0	100%
brewit_api\utils.py	28	24	14%
brewit_api\views.py	391	214	45%
manage.py	11	2	82%
-----	-----	-----	-----
TOTAL	1119	366	67%

## 9 Wirtualizacja/konteneryzacja

W ramach projektu zaimplementowano środowisko oparte na konteneryzacji przy użyciu Docker. Umożliwia to łatwe wdrażanie, skalowanie oraz zarządzanie komponentami aplikacji, minimalizując jednocześnie ryzyko problemów wynikających z różnic środowiskowych. Kluczowe komponenty aplikacji, takie jak serwer frontendowy, backendowy, baza danych oraz reverse-proxy, zostały uruchomione jako odrębne usługi w kontenerach. Aby konteneryzacja zadziałała oprócz sklonowania repozytorium należy umieścić w katalogu *certs* certyfikaty ssl o nazwach *localhost.cert* i *localhost.key*. Ponadto trzeba stworzyć plik *.env* i umieścić w nim hasło do bazy danych w postaci `POSTGRES_PASSWORD=<hasło do bazy danych>`.

### 9.1 Konfiguracja Docker Compose

Plik *docker-compose.yml* definiuje cztery główne usługi:

- reverse-proxy (NGINX),
- frontend (Flutter Web),
- backend (Django + DRF),
- db (PostgreSQL).

#### 9.1.1 Reverse Proxy (NGINX)

Usługa reverse-proxy oparta na obrazie *nginx:1.27.3* działa jako warstwa proxy obsługująca żądania HTTPS i kierująca je do odpowiednich usług (frontend/backend). Zmapowano port 443, aby host miał dostęp do systemu przez HTTPS. Szablon konfiguracji definiuje ścieżki routingu:

- `/api/` -> backend (port 8000),
- `/` -> frontend (port 8080).

Certyfikaty SSL używane przez NGINX powinny znajdować się w katalogu *certs* i powinny mieć nazwy *localhost.crt* oraz *localhost.key*.

#### 9.1.2 Frontend

Usługa frontend wykorzystuje Flutter do budowy i serwowania aplikacji frontendowej. Komenda uruchomieniowa pobiera zależności i uruchamia serwer webowy na porcie 8080.

#### 9.1.3 Backend

Usługa backend bazuje na frameworku Django z wykorzystaniem Django Rest Framework (DRF). Komenda uruchomieniowa oczekuje na postawienie bazy danych (*wait-for-it.sh*). Następnie wykonuje migracje bazy danych oraz generuje pliki statyczne. Na końcu uruchamia serwer aplikacji za pomocą Gunicorn na porcie 8000.

#### 9.1.4 Baza danych

Usługa db wykorzystuje obraz *postgres:17-alpine* jako bazę danych dla backendu. Zmiennie środowiskowe definiują szczegóły konfiguracji bazy danych, ponadto wymagane jest podanie hasła do bazy danych w pliku *.env* o czym wspomniano na początku rozdziału. Dane z bazy danych są przechowywane w wolumenie *db\_data*.

Zastosowanie konteneryzacji pozwala na efektywne zarządzanie środowiskiem aplikacji oraz umożliwia szybką iterację i wdrażanie zmian. Konfiguracja Docker Compose zapewnia spójne i skalowalne środowisko.

## 10 Bezpieczeństwo

Bezpieczeństwo jest zapewnione przez użycie szyfrowania w protokole https. Protokół https jest w naszym systemie realizowany z użyciem certyfikatu potwierdzającego tożsamość serwera.

Hasła są przechowywane w bazie danych przy użyciu haszowania z solą.

Nginx nie dopuszcza użytkowników do endpointu backend /admin. Do backendu są przekierowywane tylko endpointy zaczynające się od /api.

## 11 Podręcznik użytkownika

Nasz system zakłada istnienie trzech rodzajów użytkowników:

- administratora,
- browaru komercyjnego,
- browaru produkcyjnego.

Każdy z nich korzysta z innej części systemu i wykonuje inne akcje.

### 11.1 Administrator

Administrator to właściciel i zarządca systemu. Dostępne dla niego funkcjonalności są opisane w rozdziale 12. Podręcznik administratora.

### 11.2 Browar kontraktowy

Jest to browar, który wypożycza sprzęt od browarów produkcyjnych. Browar komercyjny uzyskuje dostęp do systemu po zaakceptowaniu prośby o rejestrację przez administratora. Browar komercyjny może przeglądać i wypożyczać urządzenia oferowane przez browary komercyjne, spełniające wybrane kryteria poprzez wybranie opcji *“Oferta browarów komercyjnych”*. Po wybraniu opcji *“Twoje rezerwacje”* można zarządzać swoimi rezerwacjami. Analogicznie można zarządzać recepturami i procesami wykonywania piwa.

### 11.3 Browar produkcyjny

Jest to browar, który wypożycza sprzęt browarom kontraktowym. Po wybraniu opcji *“Zarządzenie urządzeniami”*, użytkownik może przeglądać, edytować, dodawać i usuwać maszyny. Po wybraniu opcji *“Prośby o rezerwację od browarów kontraktowych”* użytkownik może przeglądać, akceptować i odrzucać prośby o rezerwację. Browar może zarządzać swoimi sektorami i rezerwacjami wybierając odpowiednio opcje *“Twoje sektory”* oraz *“Twoje rezerwacje”*.

## 12 Podręcznik administratora

### 12.1 Instrukcja uruchomienia systemu

Ze względu na użycie w projekcie komunikacji https, do połączenia z serwerem konieczny jest jego certyfikat.

Należy pozyskać certyfikat razem z kluczem prywatnym, w środowisku deweloperskim może być to certyfikat z podpisem własnym wygenerowany przy użyciu np. OpenSSL. Certyfikat ten należy dodać do zaufanych na komputerze, z którego łączymy się do serwera.

W środowisku produkcyjnym powinien to być certyfikat podpisany przez zaufane CA.

W środowisku produkcyjnym powinno się również zmienić w backend opcję DEBUG na False, ustawić ALLOWED\_HOSTS na swoją domenę oraz w pliku konfiguracyjnym nginx ustawić odpowiednią domenę.

System uruchamia się przy użyciu komend docker, zatem wymagane jest posiadanie systemu docker. Znajdując się w głównym katalogu należy zbudować obrazy komendą:

```
docker-compose build
```

Po zbudowaniu można uruchomić system komendą:

```
docker-compose up
```

System można wyłączyć komendą:

```
docker-compose down
```

### 12.2 Instrukcja uruchomienia komponentów

Komponenty można również uruchamiać pojedynczo przy użyciu ich własnych Dockerfile.

#### 12.2.1 Baza danych

Bazę danych uruchamia się zgodnie z dokumentacją PostgreSQL (zamieszczona w bibliografii).

#### 12.2.2 Backend

Backend uruchamia się przez przejście do folderu backend i użycie komend:

```
docker build . -t <image_name>
```

```
docker run --name <container_name> -p 8000:8000 <image_name>
```

### 12.2.3 Frontend

Frontend można zbudować poprzez przejście do folderu frontend i wpisanie komend:

```
docker build . -t <image_name>
```

```
docker run --name <container_name> -p 8080:8080 <image_name>
```

### 12.3 Zarządzanie użytkownikami i uprawnieniami

Zakładamy, że administrator ma dostęp do maszyny, na której hostowany jest system. Administrator może korzystać ze strony administratora django, dostępnej tylko na maszynie, która hostuje system (strona ta nie jest dostępna przez nginxa, który w środowisku produkcyjnym byłby jako jedyny wystawiony na świat, domyślnie można się do niej dostać tylko poprzez `http://localhost:8000/admin/`). Na stronie administratora django, można przeglądać dane w bazie danych, a także zarządzać użytkownikami. Możliwe jest zmienienie statusu konta na nieaktywne co skutkuje odcięciem dostępu do systemu. Możliwe jest także tworzenie nowych kont, w tym kont administratora.

Poza stroną administratora django, administrator ma też dostęp do funkcjonalności poprzez frontend przekazywany przez nginxa. Kiedy administrator zaloguje się na swoje konto, może zarządzać prośbami o rejestrację browarów po wybraniu opcji *“Prośby o rejestrację”*, a także przeglądać statystyki browarów po wybraniu opcji *“Statystyki”*.

## 13 Podsumowanie

### 13.1 Realizacja wymagań

#### 13.1.1 Wymagania funkcjonalne

Na dzień 10 stycznia większość wymagań funkcjonalnych została zrealizowana, do wykonania pozostały jeszcze prace związane z wyświetlaniem statystyk przez administratora.

Do prezentacji będzie też konieczne poprawienie działania niektórych funkcjonalności, dla większej czytelności i intuicyjności dla użytkownika.

#### 13.1.2 Wymagania нефunkcjonalne

Wymagania нефunkcjonalne zostały zrealizowane, zapewnione zostało bezpieczeństwo, konteneryzacja, wystarczająco krótki czas odpowiedzi i uniemożliwony został niepowołany dostęp.

### 13.2 Mocne strony systemu

Dzięki użyciu certyfikatów i https jest to rozwiązanie gotowe do dostępu publicznego, po uprzednim wygenerowaniu certyfikatów przez CA.

Konteneryzacja zapewnia niezależność od środowiska serwera.

### 13.3 Słabe strony systemu

Proces uruchomienia systemu nie jest prosty ze względu na użycie certyfikatów, jednak nie powinno to być dużym problemem, ponieważ zakładamy, że administrator ma umiejętności informatyczne.

Nie wszystkie widoki są proste do zrozumienia dla użytkownika, będzie to poprawione do prezentacji końcowej

Brak CI/CD.

### 13.4 Możliwy rozwój

Automatyzacja CI/CD.

Integracja z zewnętrznymi platformami np. systemem sprzedaży lub dostaw.

## 14 Bibliografia

The PostgreSQL Global Development Group, 2024. PostgreSQL 17.2 Documentation [online]. Adres URL: <https://www.postgresql.org/docs/17/index.html> [Data dostępu: 5 grudnia 2024].

Google, 2025. Flutter Documentation [online]. Adres URL: <https://docs.flutter.dev> [Data dostępu: 9 stycznia 2025].

Django Software Foundation, 2024. Django Documentation [online]. Adres URL: <https://docs.djangoproject.com/> [Data dostępu: 10 grudnia 2024].

Encode OSS Ltd., 2024. Django REST Framework Documentation [online]. Adres URL: <https://www.django-rest-framework.org/> [Data dostępu: 12 grudnia 2024]

Zatwierdzam dokumentację.	..... Data i podpis Mentora
---------------------------	--------------------------------