



# FITNESS

Dokumentacja projektowa PZSP2

WERSJA ...

19.01.2024

Semestr 23Z

Zespół nr ... w składzie:

Kacper Murygin

Błażej Gospodarek

Bartosz Niewiarowski

Albert Torzewski

Mentor zespołu: Robert Nowak

## Spis treści

1	Wprowadzenie .....	2
1.1	Cel projektu.....	2
1.2	Wstępna wizja projektu .....	2
2	Metodologia wytwarzania .....	2
3	Analiza wymagań .....	2
3.1	Wymagania użytkownika i biznesowe .....	3
3.2	Wymagania funkcjonalne i нефункционалне .....	2
3.3	Przypadki użycia.....	7
3.4	Potwierdzenie zgodności wymagań .....	2
4	Definicja architektury .....	14
5	Dane trwałe.....	15
5.1	Model logiczny danych .....	15
5.2	Przetwarzanie i przechowywanie danych.....	15
6	Specyfikacja analityczna i projektowa .....	16
7	Projekt standardu interfejsu użytkownika .....	17
8	Specyfikacja testów .....	22
9	<i>Wirtualizacja/konteneryzacja</i> .....	23
10	Bezpieczeństwo .....	23
11	Podręcznik użytkownika .....	23
12	Podręcznik administratora.....	26
13	Podsumowanie .....	27
14	Bibliografia .....	28

# 1 Wprowadzenie

## 1.1 Cel projektu

Celem naszego projektu jest stworzenie kompleksowego systemu mającego na celu usprawnienie działalności sieci klubów fitness. System ma wspomagać wszystkich użytkowników siłowni: klienta, trenera personalnego, portiera oraz administratora. Nasz system ma skupić się na trzech kluczowych obszarach: personalizacji treningów, śledzeniu postępów oraz efektywnym zarządzaniu klubem.

## 1.2 Wstępna wizja projektu

We wstępnej wizji naszego projektu system miał składać się z 4 oddzielnych aplikacji, oddzielnej dla każdego z najważniejszych aktorów systemowych: klienta, trenera personalnego, portiera oraz administratora. Aplikacje miałyby być zrealizowane jako aplikacje webowe, za logikę odpowiadałby serwer, który komunikowałby się ze wspólną dla wszystkich aplikacji bazą danych. Na potrzeby wstępnej implementacji systemu nie będziemy integrować systemu z zewnętrznymi urządzeniami pomiarowymi, na potrzebę systemu dane wymagane do analizy postępów użytkownika będą generowane po stronie serwera aplikacji.

# 2 Metodologia wytwarzania

Organizując pracę naszego zespołu staraliśmy się zaczerpnąć wzorce z metodologii SCRUM. Dzieliлиśmy naszą pracę na pomniejsze zadania, które zapisywaliśmy w backlogu. W celu uporządkowania naszej pracy korzystaliśmy z GitHub Issue gdzie zapisywaliśmy i przypisywaliśmy zadania na dany okres. Nowe funkcjonalności realizowaliśmy na dedykowanych temu gałęziach kodu, przed połączeniem danej gałęzi do gałęzi głównej staraliśmy się przeprowadzać code review.

W ramach spotkań zespołu planowaliśmy prace naszej całej 4, planowany rozkład prac wyglądał w następujący sposób:

### **Termin odesłania dokumentacji do oceny ekspertów 01.12.2023**

- projekt BD
- projekt UI/UX - z wykorzystaniem Figma
- określenie Architektury
- dokumentacja do ekspertów

### **Prezentacja połówkowa 14.12.2023:**

- Aplikacja klienta

### **Do dnia 12.01.2024:**

- Pozostałe aplikacje: portiera, trenera oraz administratora

### **Prezentacja końcowa 25.01.2024:**

- Kompletna dokumentacja, testy

# 3 Analiza wymagań

### 3.1 Wymagania użytkownika i biznesowe

Po rozmowach z właścicielem projektu przeprowadziliśmy analizę wymagań zgodnie z podejściem MoSCoW. Wymagania zostały zatwierdzone przez właściciela projektu.

#### **Must have**

Aplikacja Klienta:

- Autoryzacja i autentykacja użytkownika
- Możliwość umówienia treningu z trenerem personalnym
- Wyświetlanie kalendarza z planem treningu
- Wybór planu subskrypcja, uwzględniając podział na kilka typów abonamentów
- Wyświetlanie planu ćwiczeń na dany trening z uwzględnieniem ilości powtórzeń
- Obserwacja postępów użytkownika na podstawie gromadzonych danych z wykonywanych ćwiczeń oraz treningów

Aplikacja Trenera:

- Wyświetlanie kalendarza z planem treningu
- Dodawanie komentarzy pod ćwiczeniami w ramach informacji zwrotnej do klienta
- Ustalanie planu ćwiczeń klienta

Aplikacja Portiera:

- Rejestracja wejść i wyjść
- Dodanie nowego użytkownika lub usunięcie użytkownika
- Sprawdzenie informacji o koncie klienta

Aplikacja Właściciela:

- Generowanie raportów na podstawie zebranych danych:
- Statystyki wejść
- Statystyki zajętości poszczególnych maszyn

Baza danych:

- Baza będzie zaprojektowana w modelu relacyjnym
- Wszystkie 4 aplikacje będą korzystać z jednej bazy
- Baza będzie przechowywać informacje o wszystkich użytkownikach 4 aplikacji, obiektach siłowni, maszynach znajdujących się na siłowni
- Baza będzie przechowywać informacje o treningach a w tym pojedynczych seriach ćwiczeń wykonywanych przez klientów.

Inne:

- Skrypty symulujące odbyty trening z poziomu bazy danych, uzupełnienie danych dotyczących poszczególnych serii oraz pojedynczych powtórzeń. Docelowo dane te mają być gromadzone na podstawie zebranych nagrań z kamer i odczytów z czujników umieszczonych na siłowni, jednak we wstępnej implementacji aplikacji część hardwerowa zostanie pominięta.

#### **Should have**

Aplikacja Klienta:

Aplikacja Trenera:

- Możliwość odwołania treningu z klientem
- Deklarowanie dyspozycyjności

Aplikacja Portiera:

- Wgląd do kalendarza wszystkich trenerów
- Sprawdzenie ilości osób aktualnie przebywających na siłowni oraz liczbę ilości wolnych miejsc
- Obsługa płatności na koncie klienta, modyfikacja części danych użytkownika

Aplikacja Właściciela:

- Szczegółowe zestawienie finansowe danej siłowni

### Could have

Aplikacja Klienta:

- Sprawdzenie bilansu konta, nadchodzących płatności
- Możliwość zakupu planu treningowego
- Informacja o przydzielonej szafce

Aplikacja Właściciela:

- Zestawienie dwóch lub więcej obiektów, z uwzględnieniem porównania wyników finansowych oraz zajętości siłowni.
- Analiza obecności klientów w czasie
- Statystyki zajętości szafek

Aplikacja Portiera:

### Won't have

Aplikacja Klienta:

- Wersja mobilna aplikacji dla klienta
- Podłączenie API do płatności
- AI do analizowania ćwiczeń

## 3.2 Wymagania funkcjonalne i нефunkcjonalne

### Zidentyfikowani aktorzy:

- **Użytkownik Siłowni:** Użytkownik siłowni to osoba korzystająca z usług klubu fitness. Poszukuje spersonalizowanych planów treningowych, monitoruje postępy i korzysta z aplikacji mobilnej do łatwego śledzenia historii ćwiczeń. Jego doświadczenie ma być indywidualne i zoptymalizowane pod kątem osiągnięcia fitnessowych celów.
- **Trener Personalny:** Trener personalny to profesjonalista współpracujący z klientami klubu fitness. Jego zadaniem jest śledzenie postępów podopiecznych, udzielanie wskazówek oraz dostosowywanie planów treningowych. Dzięki narzędziom systemu ma efektywnie zarządzać sesjami treningowymi i utrzymywać stały kontakt z klientami.
- **Portier:** Portier pełni kluczową rolę w zarządzaniu fizycznym dostępem do klubu. Jego obowiązki obejmują kontrolę obecności klientów, obsługę systemu rezerwacji, a także udzielanie informacji

o klubie. Portier korzysta z powiadomień w systemie, aby skutecznie koordynować ruch i informować klientów o istotnych kwestiach.

- **Administrator Siłowni:** Administrator siłowni jest odpowiedzialny za strategiczne i operacyjne zarządzanie klubem fitness. Monitoruje statystyki aktywności klientów, zarządza personelem, planuje promocje i analizuje raporty finansowe. Jego celem jest efektywne i innowacyjne prowadzenie klubu, dostosowane do potrzeb klientów i wymagań rynkowych.

## User Stories:

### Użytkownik Siłowni:

1. Jako użytkownik serwisu, chcę zalogować się do serwisu, aby uzyskać dostęp do szczegółowych informacji o moim koncie.  
Kryteria akceptacji:
  - użytkownik może zalogować się do serwisu po wpisaniu poprawnych danych
2. Jako użytkownik serwisu, chcę mieć możliwość wyboru rodzaju karnetu, aby dopasować optymalny pakiet do moich indywidualnych potrzeb.  
Kryteria akceptacji:
  - użytkownik widzi dostępne pakiety wraz ze szczegółowymi informacjami o nich
  - użytkownik może zmienić wybrany pakiet
3. Jako użytkownik siłowni, chcę mieć dostęp do raportów analizujących mój postęp, aby na bieżąco nadzorować mój rozwój na przestrzeni czasu.  
Kryteria akceptacji:
  - użytkownik ma dostęp wygenerowane raporty
4. Jako użytkownik siłowni chcę zamówić trening z trenerem personalnym, aby móc rozwijać się pod okiem profesjonalisty.  
Kryteria akceptacji:
  - użytkownik ma dostęp do panelu zamawiania treningu
  - po uzupełnieniu danych dot. treningu informacja ta jest zapisywana w bazie
5. Jako użytkownik chcę mieć wgląd do mojego planu treningowego, aby móc samodzielnie realizować trening zgodnie z wyznaczonym przez trenera planem.  
Kryteria akceptacji:
  - użytkownik ma dostęp do panelu z planem na dany trening
  - dane dotyczące treningu wyświetlają się w odpowiedni sposób

### Trener Personalny:

1. Jako trener personalny chcę sprawdzić plan tygodnia, aby nie przegapić żadnego zaplanowanego treningu z podopiecznym.  
Kryteria akceptacji:
  - trener ma dostęp do panelu z planem z zaplanowanymi sesjami treningowymi

2. Jako trener personalny chcę ustalać plany treningowe moich podopiecznych, aby zadbać o jakość ich treningu i dostosować plan do ich potrzeb.  
Kryteria akceptacji:
  - trener ma możliwość dodawania ćwiczeń do treningu, po ich zatwierdzeniu klient powinien widzieć poczynione zmiany
3. Jako trener personalny chcę móc odwołać zaplanowany trening, aby w razie konieczności móc poinformować klienta o tym, że dany trening się nie odbędzie.  
Kryteria akceptacji:
  - w widoku zaplanowanych treningu powinna być opcja odwołania treningu
  - po odwołaniu treningu klient i trener nie powinni widzieć go w swoich kalendarzach
4. Jako trener personalny chcę dodać komentarz dotyczący treningu podopiecznego, aby podopieczny otrzymał wskazówki co powinien poprawić w swoim treningu.  
Kryteria akceptacji:
  - trener ma możliwość dodawania komentarzy treningu, po ich zatwierdzeniu klient powinien widzieć wprowadzone komentarze
5. Jako trener personalny chcę mieć dostęp do informacji o postępie moich podopiecznych, aby na bieżąco monitorować ich postępy na siłowni.  
Kryteria akceptacji:
  - trener ma dostęp do danych o postępach swoich podopiecznych

#### **Portier:**

1. Jako portier chcę rejestrować wejście klienta na siłownię, aby prowadzić dokładne statystyki dotyczące osób uczęszczających na dany obiekt.  
Kryteria akceptacji:
  - portier może zarejestrować wejście klienta, wejście jest widoczne w bazie danych
2. Jako portier chcę dodać nowego użytkownika do systemu, aby w ten sposób poszerzać bazę klientów naszej sieci klubów fitness.  
Kryteria akceptacji:
  - portier ma dostęp do panelu rejestracji użytkownika
  - po wprowadzeniu poprawnych danych i zatwierdzeniu rejestracji nowy użytkownik posiada konto zapisane w bazie danych
3. Jako portier chcę móc usunąć użytkownika z systemu, aby w ten sposób móc realizować ewentualne życzenie klienta.  
Kryteria akceptacji:
  - portier ma dostęp do panelu dot. danych użytkownika
  - usunięcie użytkownika jest możliwe, po usunięciu zmiany są widoczne w bazie danych
4. Jako portier chcę móc modyfikować poszczególne dane klientów, aby w ten sposób pomagać klientom, którzy nie są tego w stanie sami wykonać.  
Kryteria akceptacji:
  - portier ma dostęp do panelu dot. danych użytkownika

- modyfikacja danych użytkownika jest możliwa, po zatwierdzeniu zmiany są widoczne w bazie danych
- 5. Jako portier chcę mieć dostęp do części informacji o kliencie, aby móc udzielać mu kluczowych informacji o statusie jego konta.  
Kryteria akceptacji:
  - portier ma dostęp do panelu dot. danych użytkownika

#### **Administrator siłowni:**

1. Jako administrator siłowni chcę otrzymać raporty finansowe o zarządzanym obiekcie, aby zapewnić jak najlepsze wyniki finansowe siłowni.  
Kryteria akceptacji:
  - administrator ma dostęp do panelu administratora
  - administrator może wyświetlać raporty finansowe
2. Jako administrator siłowni chcę otrzymać raporty zajętości sprzętu, aby w razie konieczności wiedzieć jakie elementy należy dokupić.  
Kryteria akceptacji:
  - administrator ma dostęp do panelu administratora
  - administrator może wyświetlać raporty dot. Zajętości sprzętu
3. Jako administrator siłowni chcę otrzymywać raporty wejść/wyjść, aby sprawdzić zajętość siłowni oraz na podstawie tych danych dobrać optymalne godziny otwarcia oraz dobór obsady personelu.  
Kryteria akceptacji:
  - administrator ma dostęp do panelu administratora
  - administrator może wyświetlać raporty dot. wejść i wyjść na siłownię
4. Jako administrator siłowni chcę decydować o cenach pakietów oferowanych przez siłownię, aby móc wdrażać kampanie promocyjne takie jak np. czasowe obniżki cen pakietów.  
Kryteria akceptacji:
  - administrator ma dostęp do panelu administratora
  - administrator może zmieniać ceny oferowanych pakietów

### **3.3 Przypadki użycia**

#### **A. Przypadki użycia związane z klientem:**

##### **1. Rejestracja nowego konta:**

Opis: Klient zakłada konto w aplikacji, aby móc korzystać z oferty sieci klubów fitness

Warunki wstępne: klient posiada adres email

Dodatkowy aktor: -

1. Klient otwiera okno rejestracji
2. Klient wpisuje dane wymagane do rejestracji
4. Klient przesyła dane do rejestracji
  - a. w przypadku, jeśli dane są niepoprawne wyświetlany jest komunikat, klient może poprawić wprowadzone dane i wysłać prośbę o rejestrację po raz kolejny.



- b. w przypadku poprawności wprowadzonych danych konto użytkownika zostaje zarejestrowane, wyświetlany jest widok główny aplikacji

## **2. Logowanie do konta:**

Opis: Klient loguje się do aplikacji, aby dotrzeć do szczegółowych informacji o koncie

Warunki wstępne: klient posiada aktywne konto w serwisie

Dodatkowy aktor: -

1. Klient otwiera okno logowania w aplikacji
2. Klient wpisuje adres e-mail
3. Klient wpisuje hasło
4. Następuje próba logowania
  - a. jeśli adres e-mail i hasło są poprawne operacja kończy się sukcesem
  - b. jeśli adres e-mail i hasło są niepoprawne operacja kończy się niepowodzeniem, zostaje wyświetlony komunikat, użytkownik ma szansę wprowadzić po raz kolejny dane do logowania

## **3. Wybór rodzaju subskrypcji:**

Opis: Klient wybiera jeden z oferowanych pakietów zgodnie ze swoimi preferencjami

Warunki wstępne: klient posiada aktywne konto w serwisie, klient jest zalogowany

Dodatkowy aktor: -

1. Klient otwiera aplikację
2. Klient loguje się do aplikacji
3. Klient przechodzi do ekranu wyboru pakietów
4. Klient wybiera pakiet, który go interesuje
  - a. Jeśli klient nie ma aktywnego pakietu, nowy pakiet aktywuje się kolejnego dnia od zakupu, zwracana jest informacja do klienta.
  - b. Jeśli klient posiada aktywny pakiet, nowy pakiet aktywuje się po wygaśnięciu aktualnego pakietu, zwracana jest informacja do klienta.
5. Klient jest informowany o pozytywnym zakończeniu procesu.

## **4. Umówienie treningu z trenerem:**

Opis: Klient decyduje się na dodatkową usługę jaką jest trening pod okiem trenera personalnego

Warunki wstępne: klient posiada aktywne konto w serwisie, klient nie zalega z żadnymi płatnościami

Dodatkowy aktor: trener personalny

1. Klient otwiera aplikację
2. Klient loguje się do aplikacji
3. Klient przechodzi do ekranu rezerwacji treningu
4. Klient wybiera interesującą go datę oraz godzinę.
5. Wyświetlają się trenerzy, którzy są dostępni w tym terminie.
  - a. Jeśli w tym terminie nie ma dostępnych trenerów, zwracana jest informacja do klienta o niemożności odbycia wtedy treningu.
  - b. Jeśli są dostępni trenerzy, klient wybiera jednego z nich i kończy proces umawiania treningu pod okiem trenera personalnego.

## **5. Uzupełnienie danych osobowych:**

Opis: Klient chce uzupełnić lub zaktualizować swoje dane osobowe w aplikacji.

Warunki wstępne: klient ma aktywne konto w serwisie.

Dodatkowy aktor: -

1. Klient otwiera aplikację.
2. Klient loguje się do aplikacji.
3. Klient przechodzi do ekranu wyświetlającego jego dane.
4. Klient wprowadza lub aktualizuje swoje dane, takie jak imię, nazwisko, numer telefonu itp.
5. Po zapisaniu zmian klient otrzymuje potwierdzenie o udanej aktualizacji danych.

## **6. Realizacja płatności za karnet:**

Opis: Klient dokonuje płatności za karnet w celu przedłużenia lub zakupu nowego

Warunki wstępne: klient ma aktywne konto w serwisie,

klient wybrał pakiet i chce dokonać płatności.

Dodatkowy aktor: -

1. Klient otwiera aplikację.
2. Klient loguje się do aplikacji.
3. Klient przechodzi do sekcji płatności lub zakupu karnetu.
4. Klient wybiera preferowany sposób płatności (karta kredytowa, przelew bankowy itp.).
5. Klient wprowadza niezbędne dane płatnicze.
6. Po pomyślnej transakcji klient otrzymuje potwierdzenie płatności oraz informację o nowym okresie ważności karnetu.

## **7. Rezygnacja z karnetu:**

Opis: Klient chce zrezygnować z aktualnego karnetu.

Warunki wstępne: klient ma aktywny karnet.

Dodatkowy aktor: -

1. Klient otwiera aplikację.
2. Klient loguje się do aplikacji.
3. Klient przechodzi do sekcji zarządzania karnetem.
4. Klient wybiera opcję "Zrezygnuj z karnetu".
5. Klient podaje powód rezygnacji (opcjonalnie).
6. Po potwierdzeniu rezygnacji klient otrzymuje informację o zakończeniu obowiązywania karnetu.

## **B. Przypadki użycia związane z trenerem:**

### **1. Logowanie do konta:**

Opis: Trener loguje się do aplikacji, aby dotrzeć do szczegółowych informacji o koncie

Warunki wstępne: trener posiada aktywne konto w serwisie, jego konto ma uprawnienia trenera

Dodatkowy aktor: -

1. Trener otwiera okno logowania w aplikacji
2. Trener wpisuje adres e-mail

3. Trener wpisuje hasło
4. Następuje próba logowania
  - a. jeśli adres e-mail i hasło są poprawne operacja kończy się sukcesem
  - b. jeśli adres e-mail i hasło są niepoprawne operacja kończy się niepowodzeniem, zostaje wyświetlony komunikat, użytkownik ma szansę wprowadzić po raz kolejny dane do logowania

## **2. Sprawdzenie planu zajęć na tydzień przez trenera:**

Opis: Trener chce przeglądać plan zajęć na dany tydzień, aby lepiej zarządzać swoim czasem i dostosować treningi klientów.

Warunki wstępne: Trener ma dostęp do panelu trenera w aplikacji.

Dodatkowy aktor: -

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener przechodzi do zakładki "Plan zajęć" lub "Kalendarz treningów".
3. Trener wybiera konkretny dzień do sprawdzenia w kalendarzu.
4. Trener ma dostęp do informacji o treningach zaplanowanych na dany dzień

## **3. Ustalanie nowego planu treningowego przez trenera:**

Opis: Trener chce dostosować plan treningowy klienta lub stworzyć nowy plan na podstawie jego celów i postępów.

Warunki wstępne: Trener ma dostęp do panelu trenera w aplikacji, klient ma aktywne konto w serwisie.

Dodatkowy aktor: klient

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener wybiera klienta, dla którego chce ustalić nowy plan treningowy.
3. Trener analizuje aktualne postępy klienta, uwzględniając jego cele i ewentualne zmiany w kondycji fizycznej.
4. Trener tworzy nowy plan treningowy, określając ćwiczenia, ilość serii, powtórzeń, obciążenia itp.
5. Trener przypisuje nowy plan treningowy do konta klienta.
6. Klient otrzymuje powiadomienie o dostępie do nowego planu treningowego.

## **4. Dodanie komentarza do treningu przez trenera:**

Opis: Trener chce dodać komentarz do wykonanego przez klienta treningu w celu udzielenia informacji zwrotnej lub dostosowania planu.

Warunki wstępne: Trener ma dostęp do panelu trenera w aplikacji, klient ma aktywny plan treningowy.

Dodatkowy aktor: klienta

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener wybiera konkretny trening wykonany przez klienta.
3. Trener dodaje komentarz, podając informacje zwrotne, zalecenia lub ewentualne modyfikacje do planu treningowego.
4. Klient otrzymuje powiadomienie o dodaniu komentarza do jego treningu.

## **5. Wgląd trenera w postępy klienta:**

Opis: Trener chce sprawdzić postępy swojego klienta, aby dostosować plan treningowy lub udzielić lepszych wskazówek.

Warunki wstępne: Trener ma dostęp do panelu trenera w aplikacji, klient ma aktywny plan treningowy.

Dodatkowy aktor: klient

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener wybiera klienta i przechodzi do zakładki "Postępy" lub "Statystyki".
3. Trener ma dostęp do danych o postępach klienta

## **6. Anulowanie treningu przez trenera:**

Opis: Trener musi anulować wcześniej umówiony trening z klientem.

Warunki wstępne: Trener ma dostęp do panelu trenera w aplikacji, trening został wcześniej umówiony z klientem.

Dodatkowy aktor: klient

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener przechodzi do kalendarza lub listy umówionych treningów.
3. Trener wybiera konkretny trening, który chce anulować.
4. Trener wprowadza uzasadnienie anulowania treningu.
5. Klient otrzymuje powiadomienie o anulowaniu treningu, a w razie potrzeby trener i klient ustalają nowy termin.

## **C. Przypadki użycia związane z portierem:**

### **1. Logowanie do konta:**

Opis: Portier loguje się do aplikacji, aby dotrzeć do szczegółowych informacji o koncie

Warunki wstępne: klient posiada aktywne konto w serwisie

1. Portier otwiera aplikację
2. Portier wpisuje adres e-mail
3. Portier wpisuje hasło
4. Następuje próba logowania
  - a. jeśli adres e-mail i hasło są poprawne operacja kończy się sukcesem
  - b. jeśli adres e-mail i hasło są niepoprawne operacja kończy się niepowodzeniem, zostaje wyświetlony komunikat, użytkownik ma szansę wprowadzić po raz kolejny dane do logowania

### **2. Rejestracja wejścia użytkownika przez portiera:**

Opis: Portier chce zarejestrować wejście użytkownika na siłownię.

Warunki wstępne: Portier ma dostęp do panelu rejestracji wejść, użytkownik ma aktywny karnet na siłownię.

Dodatkowy aktor: klient

1. Portier otwiera system rejestracji wejść na komputerze.
2. Portier wybiera opcję "Rejestracja wejścia".
3. Portier wyszukuje użytkownika w systemie i rejestruje jego wejście.
4. System potwierdza poprawne zarejestrowanie wejścia użytkownika.

### 3. Dodanie nowego użytkownika przez portiera:

Opis: Portier chce dodać nowego użytkownika do systemu.

Warunki wstępne: Portier ma dostęp do panelu administracyjnego systemu, nowy użytkownik chce dołączyć do siłowni.

Dodatkowy aktor: klient

1. Portier otwiera panel administracyjny systemu.
2. Portier wybiera opcję "Dodaj nowego użytkownika".
3. Portier wprowadza dane nowego użytkownika, takie jak imię, nazwisko, numer telefonu, adres e-mail itp.
4. System generuje unikalny identyfikator użytkownika (np. numer karnetu) oraz nadaje mu dostęp do siłowni.

### 4. Usunięcie użytkownika przez portiera:

Opis: Portier chce usunąć użytkownika z systemu (np. po zakończeniu karnetu lub na jego żądanie).

Warunki wstępne: Portier ma dostęp do panelu administracyjnego systemu, użytkownik, którego chcemy usunąć, nie ma aktywnego karnetu.

Dodatkowy aktor: klient

1. Portier otwiera panel administracyjny systemu
2. Portier wyszukuje użytkownika, którego chce usunąć.
3. Portier przechodzi do panelu danych użytkownika
4. Portier wybiera opcję "Usuń użytkownika".
5. System potwierdza usunięcie użytkownika z bazy danych.

### 5. Obsługa płatności na miejscu na siłowni przez portiera:

Opis: Portier chce przyjąć płatność od użytkownika na miejscu w przypadku zakupu karnetu lub dodatkowych usług.

Warunki wstępne: Portier ma dostęp do panelu płatności, użytkownik chce dokonać płatności na miejscu.

Dodatkowy aktor: klient

1. Portier otwiera panel płatności.
2. Portier wybiera opcję "Nowa płatność".
3. Portier wprowadza dane transakcji, takie jak kwota, wybrany sposób płatności.

### 6. Dostęp do informacji o danym użytkowniku przez portiera:

Opis: Portier chce sprawdzić informacje dotyczące danego użytkownika, takie jak status karnetu, dane kontaktowe.

Warunki wstępne: Portier ma dostęp do panelu informacyjnego, użytkownik jest na siłowni. obecny

Dodatkowy aktor: klient

1. Portier otwiera panel administracyjny systemu.
2. Portier wyszukuje użytkownika w systemie.

3. System wyświetla informacje o danym użytkowniku, takie jak status karnetu, dane kontaktowe, aktywne usługi.

#### **7. Modyfikacja części danych użytkownika przez portiera:**

Opis: Portier chce zaktualizować część danych użytkownika, na przykład zmienić numer telefonu.

Warunki wstępne: Portier ma dostęp do panelu administracyjnego systemu, użytkownik zgłosił potrzebę aktualizacji danych.

Dodatkowy aktor: klient

1. Portier otwiera panel administracyjny systemu.
2. Portier wyszukuje użytkownika, którego dane chce zaktualizować.
3. Portier przechodzi do widoku danych użytkownika
4. Portier wprowadza nowe dane lub modyfikuje istniejące.
5. System potwierdza aktualizację danych użytkownika.

#### **D. Przypadki użycia związane z administratorem:**

##### **1. Sprawdzenie wygenerowanych raportów przez administratora siłowni:**

Opis: Administrator siłowni chce sprawdzić wygenerowane raporty dotyczące aktywności, płatności lub innych istotnych danych.

Warunki wstępne: Administrator ma dostęp do systemu raportowania siłowni.

Dodatkowy aktor: klient

1. Administrator loguje się do panelu administracyjnego systemu.
2. Administrator przechodzi do sekcji raportów.
3. Administrator wybiera interesujący go rodzaj raportu, np. raport aktywności klientów, raport płatności, raport treningów itp.
4. System generuje raport z odpowiednimi danymi.

##### **2. Wprowadzenie zmian cen przez administratora siłowni:**

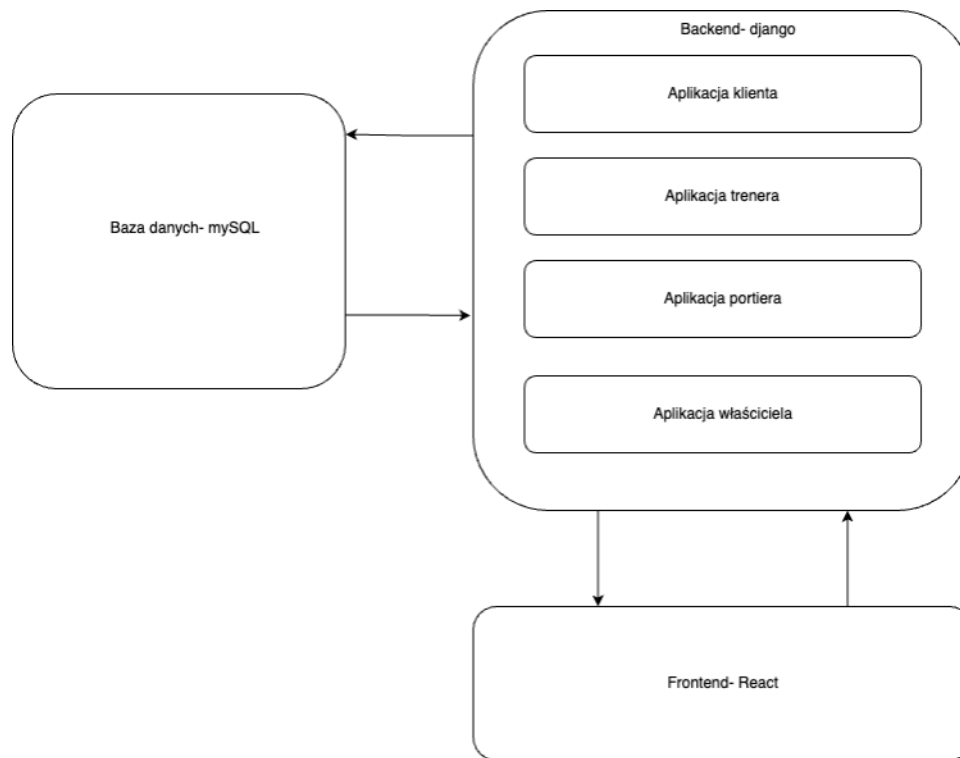
Opis: Administrator siłowni chce wprowadzić zmiany w cenach oferowanych karnetów lub usług.

Warunki wstępne: Administrator ma dostęp do panelu administracyjnego systemu.

1. Administrator loguje się do panelu administracyjnego systemu.
2. Administrator przechodzi do sekcji zarządzania cenami.
3. Administrator wybiera kategorie cenowe, które chce zmodyfikować (np. karnety, treningi z trenerem, usługi dodatkowe).
4. Administrator wprowadza nowe ceny lub modyfikuje istniejące, uwzględniając ewentualne promocje lub zniżki.
5. System zapisuje wprowadzone zmiany i aktualizuje informacje dla klientów.

## 4 Definicja architektury

Nasz backend zostanie w pełni wdrożony w języku Python, korzystając z popularnego rozwiązania jakim jest framework Django. Z kolei frontend, oparty na frameworku React, będzie komunikować się z backendem poprzez API. Dla efektywnego przechowywania danych, wybieramy bazę danych MySQL, która, analogicznie do naszej aplikacji, zostanie uruchomiona w kontenerze Dockera. Taka architektura pozwala nam skoncentrować się na efektywnym tworzeniu aplikacji, oferując przy tym skalowalność i łatwość zarządzania całym środowiskiem.



### Backend:

Backendowa strona naszego projektu zostanie napisana z użyciem Django, a dokładnie skupimy się na użyciu Django Rest Framework, aby udostępniać API, z którego będzie korzystał frontend.

Backend będzie składał się z 4 aplikacji, po jednej dla klienta siłowni, trenera, portiera i właściciela.

Jeżeli chodzi o wzorce projektowe, to jednym z wykorzystywanych przez ten framework jest wzorec Model-Serializer-View- w naszej aplikacji skorzystamy z:

- Model- do reprezentacji struktury danych w bazie danych
- Serializator- do konwertowania danych modelu do formatu JSON
- Widok- do przetwarzania żądań http

Całość backendu będzie umieszczona w jednym kontenerze Dockera.

## Frontend:

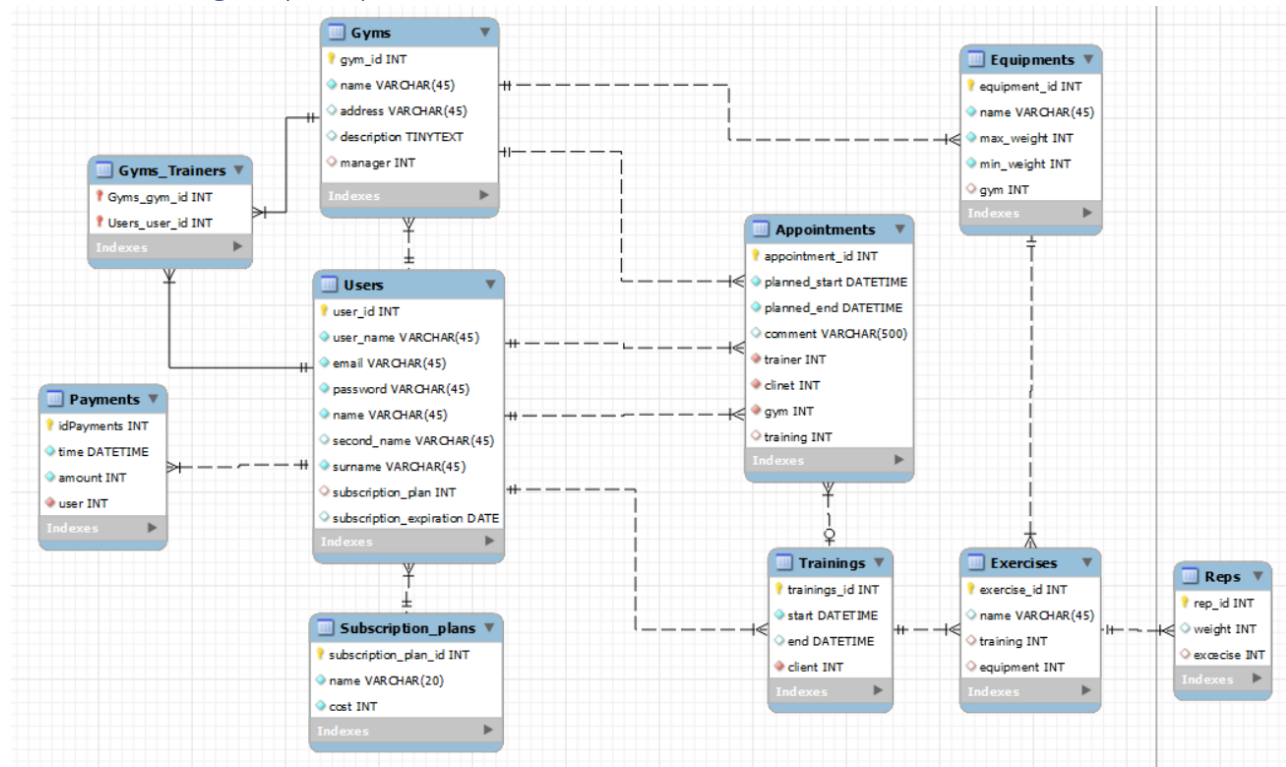
Frontend naszej aplikacji został opracowany przy użyciu popularnego frameworka jakim jest React. Wykorzystanie tej technologii umożliwiło stworzenie przyjaznej i intuicyjnej dla użytkownika warstwy prezentacji naszej aplikacji. Kod został zgodnie ze sztuką podzielony na komponenty, tak aby każdy komponent posiadał jedną kluczową funkcję. W finalnej wersji naszego projektu powstała jedna aplikacja odpowiadająca za warstwę prezentacji, po zalogowaniu użytkownik otrzymuje dostęp do paneli które wynikają z jego uprawnień, klient siłowni, nie dostanie się do panela administratora. Dodatkowo w kodzie zostały umieszczone odpowiednie komentarze, tak aby w przyszłości osoby rozwijające nasz produkt mogły w łatwiejszy i szybszy sposób zrozumieć działanie kodu stworzonego przez nas.

## Baza danych:

Baza danych zostanie zaimplementowana przy użyciu relacyjnej bazy danych MySQL, działającej w osobnym kontenerze dockera.

## 5 Dane trwałe

### 5.1 Model logiczny danych



### 5.2 Przetwarzanie i przechowywanie danych

Reprezentacja danych trwałych: relacyjna baza danych – MySQL

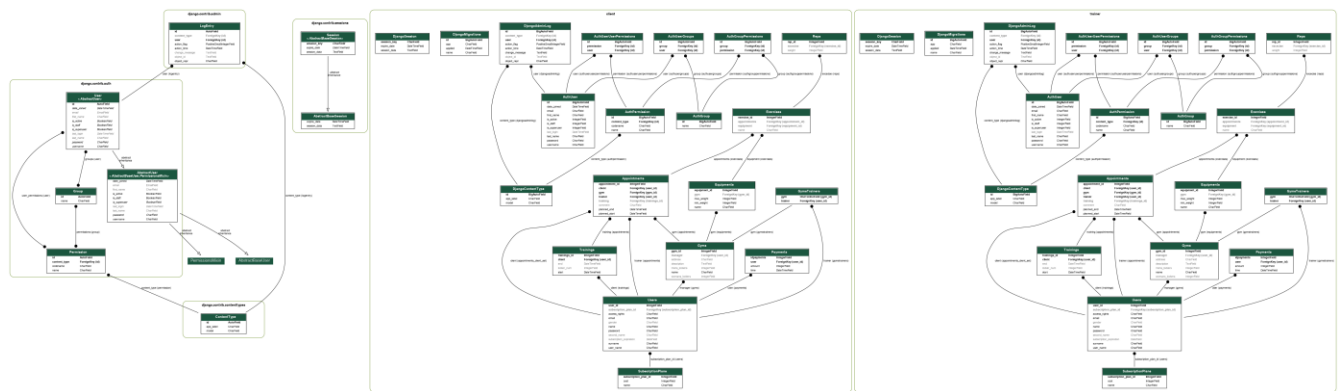


W bazie danych zostaną wykorzystane tylko i wyłącznie ograniczenia sprawdzające poprawność umieszczanych danych jak np. appointment z klientem może zostać umówiony z trenerem tylko i wyłącznie na siłowni, na której ten trener pracuje.

**Odnosnik do repozytorium:**

**Nasz projekt został opracowany przy użyciu następujących technologii:**

- Diagram klas:**



- Liczba plików: 112, w tym:
  - Python: 27
  - JavaScript: 42
  - CSS: 27
  - SQL: 2
  - YAML: 1
  - Dockerfile: 3
- Liczba linii kodu:
  - Python: 1301
  - JavaScript: 2324

- CSS: 1271
- SQL: 424
- YAML: 46
- Dockerfile: 21
- Liczba testów jednostkowych:

## 6 Projekt standardu interfejsu użytkownika

Projekt standardu użytkownika wykonaliśmy z użyciem narzędzia FIGMA:

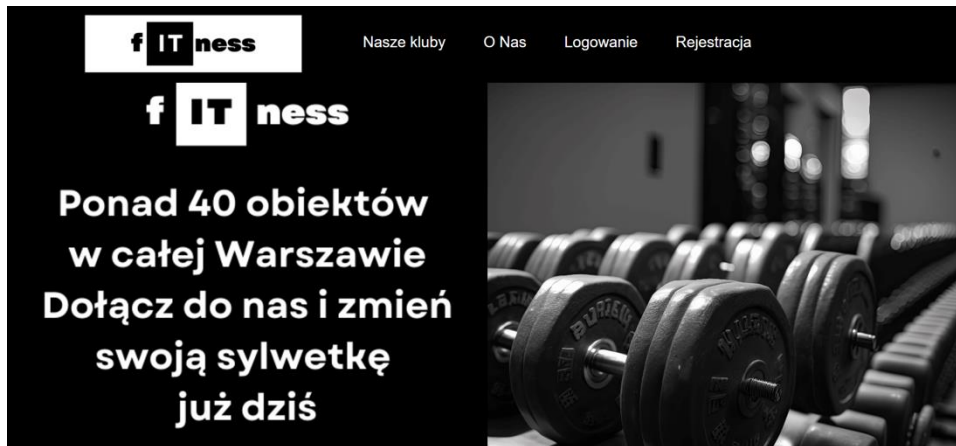
Odknośnik do opracowanego prototypu:

<https://www.figma.com/file/5Mf9QrAGmxpNyl5sF2Oqo3/fITness?type=design&node-id=0%3A1&mode=design&t=iFKJoeT1yYce4JAM-1fac>

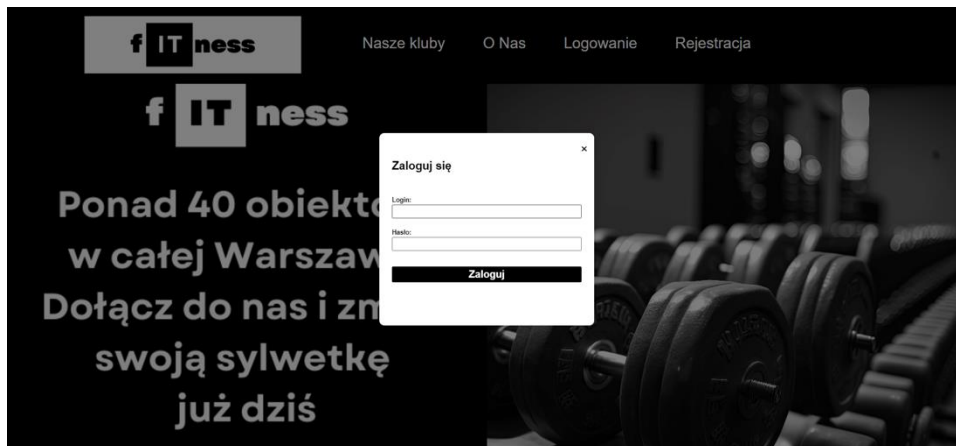
Na podstawie przygotowanego prototypu opracowaliśmy interfejs użytkownika w finalnej wersji aplikacji, oto wyniki naszych prac:

### Strona główna aplikacji

- Ekran startowy



- Ekran logowania



- Ekran rejestracji

**fITness**

Nasze Kluby O Nas Logowanie Rejestracja

**fITness**

Ponad 40 obiektów w całej Warszawie. Dołącz do nas i znajdź swoją sylwetkę już dziś.

**Zarejestruj się już dziś**

Nazwa Użytkownika:

Hasło:

Powtórz Hasło:

Podaj swoje dane

Email:

Imię:

Drugie Imię:

Nazwisko:

Płeć:

**Zarejestruj**

## Moduł Klienta

- Widok kalendarza z zaplanowanymi treningami

**Twój Kalendarz**

Sprawdź plan na ten tydzień

17 Jan 24

**Trening na Siłowni**

Cardio Warm-up  
10 minut

Weighted Squats  
10 minut

Running Intervals  
10 minut

Battle Ropes  
10 minut

Rowing Machine  
10 minut

- Widok kalendarza z możliwością zaplanowania treningu

Sprawdź plan na ten tydzień

JAN 2024						
MON	TUE	WED	THU	FRI	SAT	SUN
15	16	17	18	19	20	21
poprzedni tydzień		3			następny tydzień	

Sun 21 Jan 24

Wybierz Godzinę Startu:

Wybierz Godzinę Zakończenia:

Wybierz Trenera:

Wybierz Siłownię:

- Widok wyboru pakietu

[Nasze kluby](#)
[O Nas](#)
[Moje Konto](#)
[Wyloguj sie](#)

[Kalendarz](#)
[Mój pakiet](#)
[Moje dane](#)
[Progres](#)

Basic

Gold

Platinum

Premium

Platinum

Cena pakietu: 180

Zrezygnuj z subskrypcji

- Widok rezygnacji z pakietu

[Nasze kluby](#)
[O Nas](#)
[Moje Konto](#)
[Wyloguj sie](#)

[Kalendarz](#)
[Mój pakiet](#)
[Moje dane](#)
[Progres](#)

Basic

Platinum

Premium

Platinum

Cena pakietu: 180

Zrezygnuj z subskrypcji

Czy jesteś pewien że chcesz zrezygnować z subskrypcji?

Tak

- Widok danych użytkownika

[Kalendarz](#)
[Mój pakiet](#)
[Moje dane](#)
[Progres](#)

Nazwa użytkownika:

example\_user

Id użytkownika:

1

Imię:

Johnee

Dругie imię:

Middle

Nazwisko:

Doe

Pln...

- Widok z wyświetlaniem raportów dotyczących użytkownika

[Kalendarz](#)
[Mój pakiet](#)
[Moje dane](#)
[Progres](#)

## Twój progress na wybrany miesiąc

Razem na siłowni przebywałeś/aś przez 1.5011111111111111 godzin!

W tym czasie odbyłeś/aś 2 treningów


W tym czasie odbyłeś/aś 4 zajęć z trenerami osobistym

Twoi ulubieni trenerzy to: Byczek

Styczeń 2024

## Moduł Trenera

- Widok planu treningów trenera



[Nasze kluby](#)
[O Nas](#)
[Moje Konto](#)
[Wyloguj sie](#)

Przejdź do: Trener

### Twój Kalendarz

Sprawdź plan na ten tydzień

JAN 2024						
pon	wt	śro	czw	pt	sob	nied
15	16	17	18	19	20	21
poprzedni tydzień		3			następny tydzień	

Współ 17 Jan 24

**Trening ID: 1**

Start: 18:00

Koniec: 19:00

Historia: Byczek Trening

Stanowisko

**Trening ID: 6**

Start: 18:00

Koniec: 19:00

Historia: Byczek

Stanowisko

**Skontaktuj się z nami**

Tel: +48 123 456 789 Adres: ul. Przykładowa 123, 00-000 Warszawa

E-mail: f IT ness@gmail.com

- Widok szczegółów treningu

fITness

Nasze klubyO NasMoje KontoWyloguj sięPrzejdź do:Trener

Twój Kalendarz

Sprawdź plan na ten tydzień

15161718192021

3

17 Jan 24

Trening ID: 1

Start: 10:00

Koniec: 10:00

Komentarz: Baw się dobrze trening

Zakończ

Trening na Siłowni

Cardio Warm-up

Musi być

Weighted Squats

Musi być

Running Intervals

Musi być

- Widok z możliwością dodania ćwiczenia bądź odwołania treningu

fITness

Nasze klubyO NasMoje KontoWyloguj sięPrzejdź do:Trener

Twój Kalendarz

Sprawdź plan na ten tydzień

15161718192021

3

17 Jan 24

Trening ID: 703742

Start: 10:00

Koniec: 10:00

Komentarz:

Zakończ

Trening na Siłowni

Odeślij trening

Odeślij ćwiczenie

Skontaktuj się z nami

Telefon: +48 123 456 789 | Adres: ul. Przykładowa 123, 00-000 Warszawa

Email: [kontakt@fittestness.pl](mailto:kontakt@fittestness.pl)

Moduł portiera:

- Widok z listą użytkowników

fITness

Nasze klubyO NasMoje KontoWyloguj sięPrzejdź do:Portier

Dodaj nowego użytkownika

ID	Imię	Nazwisko	Email	Zarejestruj Wejście	Zarejestruj Wyjście
1	Johnee	Doe	example@email.com	Zarejestruj	Zarejestruj
2	Lorem	Mone	impsum@email.com	Zarejestruj	Zarejestruj
3	Kanye	East	rap_forever@email.com	Zarejestruj	Zarejestruj
4	Stefan	Byczek	byku@email.com	Zarejestruj	Zarejestruj
5	Ewelina	Kwas	efclunia@email.com	Zarejestruj	Zarejestruj
6	Bogdan	Kowalski	egirl.uwu@email.com	Zarejestruj	Zarejestruj
7	Aga	Zaradza	szeftroku@email.com	Zarejestruj	Zarejestruj
117183	piotr	piotr	piotr	Zarejestruj	Zarejestruj

Skontaktuj się z nami

- Widok szczegółów o koncie użytkownika

**f IT ness**

Nasze kluby   O Nas   Moje Konto   Wyloguj się   Przejdź do: Portier

Profil użytkownika: 2

Nazwa użytkownika:

Id użytkownika:

Imię:

Drugie imię:

Nazwisko:

Płeć:

## 7 Specyfikacja testów

W ramach testów funkcjonalnych, zostały przeprowadzone następujące testy:

### Testy wspólne dla wszystkich użytkowników:

- Login\_user – logowanie użytkownika poprzez okno logowania
- Logout\_user – wylogowanie użytkownika

### Testy specyficzne dla klienta siłowni:

- Register\_user – rejestracja użytkownika w panelu rejestracji dostępnym z poziomu widoku startowego
- Change\_user\_data – samodzielna zmiana danych użytkownika
- Add\_appointment – zaplanowanie nowego treningu ze wskazaniem na wybraną datę, godzinę, trenera i siłownię
- Choose\_subscription – zmiana wybranej subskrypcji w panelu subskrypcje
- Unsubscription – rezygnacja z subskrypcji

### Testy specyficzne dla portiera:

- Register\_entry – rejestracja wejścia danego użytkownika
- Register\_exit – rejestracja wyjścia danego użytkownika
- Change\_data\_by\_portier – zmiana danych użytkownika przez portiera
- Register\_user\_by\_portier - rejestracja użytkownika w panelu rejestracji dostępnym z poziomu widoku użytkownika

### Testy specyficzne dla trenera:

- Check\_day\_plan - wyświetlenie planu treningów w klientami na dany dzień
- Add\_comment – dodanie komentarza do treningu który już się odbył
- Add\_exercise – dodanie ćwiczenia do zaplanowanego treningu
- Remove\_exercise - usunięcie ćwiczenia przed rozpoczęciem treningu
- Cancel\_training – odwołanie zaplanowanego treningu przez trenera.

## 8 Wirtualizacja/konteneryzacja

W celu efektywnego zarządzania środowiskiem i wdrażania naszej aplikacji, zastosowaliśmy konteneryzację przy użyciu technologii Docker. Każdy element naszej aplikacji jest uruchamiany w oddzielnym kontenerze, co umożliwia nam osiągnięcie izolacji, przenośności i łatwości w zarządzaniu.

### W ramach konteneryzacji przy użyciu Docker:

- Każdy komponent naszej aplikacji: baza danych, backend i frontend, jest zawarty w osobnym kontenerze Docker. To pozwala na uniezależnienie i skalowanie poszczególnych elementów niezależnie od siebie.
- Do definiowania i zarządzania wieloma kontenerami jednocześnie używamy Docker Compose.
- Wykorzystujemy mapowanie portów Docker, aby umożliwić dostęp do poszczególnych usług z poziomu hosta. Komunikacja między kontenerami jest zdefiniowana poprzez sieć Docker, co zapewnia bezpieczne i efektywne komunikowanie się komponentów.
- Aby przechowywać dane trwałe, stosujemy woluminy Docker, co pozwala nam separować dane od kontenerów. Woluminy są używane zwłaszcza dla bazy danych MySQL, gdzie przechowujemy dane użytkowników.

Schemat jak to

## 9 Bezpieczeństwo

## 10 Podręcznik użytkownika

### 1. Rejestracja nowego konta:

1. Klient otwiera okno rejestracji
2. Klient wpisuje dane wymagane do rejestracji
4. Klient przesyła dane do rejestracji
  - a. w przypadku, jeśli dane są niepoprawne wyświetlany jest komunikat, klient może poprawić wprowadzone dane i wysłać prośbę o rejestrację po raz kolejny.
  - b. w przypadku poprawności wprowadzonych danych konto użytkownika zostaje zarejestrowane, wyświetlany jest widok główny aplikacji

### 2. Logowanie do konta:

1. Klient otwiera okno logowania w aplikacji
2. Klient wpisuje adres e-mail
3. Klient wpisuje hasło
4. Następuje próba logowania
  - a. jeśli adres e-mail i hasło są poprawne operacja kończy się sukcesem
  - b. jeśli adres e-mail i hasło są niepoprawne operacja kończy się niepowodzeniem, zostaje wyświetlony komunikat, użytkownik ma szansę wprowadzić po raz kolejny dane do logowania

### 3. Wybór rodzaju subskrypcji:

1. Klient otwiera aplikację
2. Klient loguje się do aplikacji
3. Klient przechodzi do ekranu wyboru pakietów



4. Klient wybiera pakiet, który go interesuje
  - a. Jeśli klient nie ma aktywnego pakietu, nowy pakiet aktywuje się kolejnego dnia od zakupu, zwracana jest informacja do klienta.
  - b. Jeśli klient posiada aktywny pakiet, nowy pakiet aktywuje się po wygaśnięciu aktualnego pakietu, zwracana jest informacja do klienta.
5. Klient jest informowany o pozytywnym zakończeniu procesu.

#### **4. Umówienie treningu z trenerem:**

1. Klient otwiera aplikację
2. Klient loguje się do aplikacji
3. Klient przechodzi do ekranu rezerwacji treningu
4. Klient wybiera interesującą go datę oraz godzinę.
5. Wyświetlają się trenerzy, którzy są dostępni w tym terminie.
  - a. Jeśli w tym terminie nie ma dostępnych trenerów, zwracana jest informacja do klienta o niemożności odbycia wtedy treningu.
  - b. Jeśli są dostępni trenerzy, klient wybiera jednego z nich i kończy proces umawiania treningu pod okiem trenera personalnego.

#### **5. Uzupełnienie danych osobowych:**

1. Klient otwiera aplikację.
2. Klient loguje się do aplikacji.
3. Klient przechodzi do ekranu wyświetlającego jego dane.
4. Klient wprowadza lub aktualizuje swoje dane, takie jak imię, nazwisko, numer telefonu itp.
5. Po zapisaniu zmian klient otrzymuje potwierdzenie o udanej aktualizacji danych.

#### **6. Realizacja płatności za karnet:**

1. Klient otwiera aplikację.
2. Klient loguje się do aplikacji.
3. Klient przechodzi do sekcji płatności lub zakupu karnetu.
4. Klient wybiera preferowany sposób płatności (karta kredytowa, przelew bankowy itp.).
5. Klient wprowadza niezbędne dane płatnicze.
6. Po pomyślnej transakcji klient otrzymuje potwierdzenie płatności oraz informację o nowym okresie ważności karnetu.

#### **7. Rezygnacja z karnetu:**

1. Klient otwiera aplikację.
2. Klient loguje się do aplikacji.
3. Klient przechodzi do sekcji zarządzania karnetem.
4. Klient wybiera opcję "Zrezygnuj z karnetu".
5. Klient podaje powód rezygnacji (opcjonalnie).
6. Po potwierdzeniu rezygnacji klient otrzymuje informację o zakończeniu obowiązywania karnetu.

#### **8. Sprawdzenie planu zajęć na tydzień przez trenera:**

1. Trener loguje się do panelu trenera w aplikacji.

2. Trener przechodzi do zakładki "Plan zajęć" lub "Kalendarz treningów".
3. Trener wybiera konkretny dzień do sprawdzenia w kalendarzu.
4. Trener ma dostęp do informacji o treningach zaplanowanych na dany dzień

#### **9. Dodanie komentarza do treningu przez trenera:**

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener wybiera konkretny trening wykonany przez klienta.
3. Trener dodaje komentarz, podając informacje zwrotne, zalecenia lub ewentualne modyfikacje do planu treningowego.
4. Klient otrzymuje powiadomienie o dodaniu komentarza do jego treningu.

#### **10. Wgląd trenera w postępy klienta:**

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener wybiera klienta i przechodzi do zakładki "Postępy" lub "Statystyki".
3. Trener ma dostęp do danych o postępach klienta

#### **11. Anulowanie treningu przez trenera:**

1. Trener loguje się do panelu trenera w aplikacji.
2. Trener przechodzi do kalendarza lub listy umówionych treningów.
3. Trener wybiera konkretny trening, który chce anulować.
4. Trener wprowadza uzasadnienie anulowania treningu.
5. Klient otrzymuje powiadomienie o anulowaniu treningu, a w razie potrzeby trener i klient ustalają nowy termin.

#### **12. Rejestracja wejścia użytkownika przez portiera:**

Opis: Portier chce zarejestrować wejście użytkownika na siłownię.

Warunki wstępne: Portier ma dostęp do panelu rejestracji wejść, użytkownik ma aktywny karnet na siłownię.

Dodatkowy aktor: klient

1. Portier otwiera system rejestracji wejść na komputerze.
2. Portier wybiera opcję "Rejestracja wejścia".
3. Portier wyszukuje użytkownika w systemie i rejestruje jego wejście.
4. System potwierdza poprawne zarejestrowanie wejścia użytkownika.

#### **13. Dodanie nowego użytkownika przez portiera:**

1. Portier otwiera panel administracyjny systemu.
2. Portier wybiera opcję "Dodaj nowego użytkownika".
3. Portier wprowadza dane nowego użytkownika, takie jak imię, nazwisko, numer telefonu, adres e-mail itp.
4. System generuje unikalny identyfikator użytkownika (np. numer karnetu) oraz

mu nadaje dostęp do siłowni.

#### **14. Usunięcie użytkownika przez portiera:**

1. Portier otwiera panel administracyjny systemu
2. Portier wyszukuje użytkownika, którego chce usunąć.
3. Portier przechodzi do panelu danych użytkownika
4. Portier wybiera opcję "Usuń użytkownika".
5. System potwierdza usunięcie użytkownika z bazy danych.

#### **15. Dostęp do informacji o danym użytkowniku przez portiera:**

1. Portier otwiera panel administracyjny systemu.
2. Portier wyszukuje użytkownika w systemie.
3. System wyświetla informacje o danym użytkowniku, takie jak status karnetu, dane kontaktowe, aktywne usługi.

#### **16. Modyfikacja części danych użytkownika przez portiera:**

1. Portier otwiera panel administracyjny systemu.
2. Portier wyszukuje użytkownika, którego dane chce zaktualizować.
3. Portier przechodzi do widoku danych użytkownika
4. Portier wprowadza nowe dane lub modyfikuje istniejące.
5. System potwierdza aktualizację danych użytkownika.

## **11 Podręcznik administratora**

### **Budowy systemu z kodu źródłowego:**

- Administrator posiada zainstalowany Docker Desktop
- Administrator znajduje się w katalogu głównym projektu [PZSP2-2023Z-fITness-SzczurySiłowni]
- Administrator uruchamia komendę: `docker-compose up --build`

### **Instrukcja aktualizacji oprogramowania:**

- Administrator posiada zainstalowany Docker Desktop
- Administrator znajduje się w katalogu głównym projektu [PZSP2-2023Z-fITness-SzczurySiłowni]
- Administrator usuwa starą wersję kontenerów w celu ich przebudowania, uruchamiając komendę: `docker-compose down`
  - Jeżeli zmiany dotyczą modelu baz danych na początku należy wygenerować dump bazy w celu zachowania danych i umieścić go w folderze `/fITness_db/db_init`
  - Usuwanie kontenerów w przypadku zmiany bazy powinno się wykonywać z flagą `-v` usuwającą również woluminy kontenerów
- Po wprowadzeniu zmian administrator uruchamia komendę: `docker-compose up --build`

### **Instalacji i konfiguracji systemu:**

- W celu instalacji należy zainstalować Docker Desktop oraz pobrać kod źródłowy z repozytorium
- System używa automatycznej konfiguracji poprzez wykorzystanie mechanizmów Dockera, którym jest `docker-compose`

- W celu zmian konfiguracji należy w stosowny sposób edytować plik docker-compose.yml w ramach potrzeb oraz ewentualnie odpowiedni pliki Dockerfile odpowiadające z poszczególne komponenty systemu takie jak – baza danych, backend app oraz frontend app

#### **Tworzenie kopii zapasowych i odtwarzania systemu:**

- W celu odtworzenia systemu powinno się wykonać te same kroki co w przypadku aktualizacji oprogramowania
- W celu utworzenia kopii zapasowej bazy danych należy wygenerować plik wykonując komendę:  
docker exec pzp2-db mysqldump -u "nazwa\_użytkownika" -p pzp2-db > dump.sql
  - Następnie przenieść plik z kopią zapasową do bezpiecznej lokalizacji
  - Zalecane jest umieszczenie aktualnej daty w nazwie utworzonego pliku kopii zapasowej

#### **Zarządzania użytkownikami i uprawnieniami:**

- Ze względu na aktualny brak istnienia odpowiedniego panelu administratora w celu utworzenia nowego użytkownika należy wykonać odpowiednie komendy sql na tabeli Users w bazie danych, znajdującej się w kontenerze
- W celu zmiany uprawnień użytkownika należy zmienić kolumnę access\_rights w tabeli Users poprzez umieszczenie 3 literowego skrótu odpowiadającemu wybranemu poziomowi dostępu (USR - zwykły użytkownik, TRN - trener, PRT - portier, ADM - admin)

## **12 Podsumowanie**

Zakres projektu realizowanego w ramach przedmiotu PZSP2 był duży, w istocie na projekt składało się stworzenie 4 dużych komponentów aplikacji, dedykowany dla każdego aktora systemowego. W ramach pracy naszemu zespołowi udało się zrealizować większość postawionych przed nami wyzwań. Opracowaliśmy 3 z 4 głównych komponentów aplikacji webowej, każdy z nich realizuje szereg funkcjonalności, aplikacja webowa połączona jest z backendem aplikacji przy pomocy dedykowanego API, a w ramach systemu dane są przechowywane w zaprojektowanej na jej potrzeby bazy danych. Realizacja projektu umożliwiła nam nabycie nowych umiejętności:

- Lepsze poznanie technologii z którymi pracowaliśmy: React, Django, MySQL.
- Formułowanie i analiza wymagań biznesowych
- Poznanie pracy w zespole kierującym się w swoim działaniu metodyką zwinnego zarządzaniem projektem.

Jednak w ramach realizacji projektu napotkaliśmy kilka problemów, nad których rozwiązaniem pracowaliśmy wiele czasu:

- Jednym z głównych problemów była konfiguracja testów jednostkowych do backendu naszej aplikacji, mimo licznych prób naprawienia tego problemu, niestety nie udało nam się go rozwiązać.
- W tej sekcji również warto zauważyć trudność jaką była estymacja wymaganego czasu pracy na realizację poszczególnych elementów aplikacji, niestety w kilku przypadkach termin realizacji wydłużył się i nie udało się dostarczyć poszczególnych komponentów na wcześniej wyznaczony termin. Wynikało to z braku doświadczenia w estymacji złożoności pracy, które nabierzemy w trakcie pracy nad innymi projektami.

## 13 Bibliografia

Zatwierdzam dokumentację.	<hr/> <div data-bbox="1136 415 1430 457">Data i podpis Mentora</div>
---------------------------	--