



SIEĆ SIŁOWNI

Dokumentacja projektowa PZSP2

WERSJA FINALNA

25.11.2023

Semestr 23Z

Zespół nr 5 w składzie:

Magda Twarowska
Maria Polakowska
Marta Sobol
Małgorzata Kozłowska

Mentor zespołu: mgr. inż. Klara Borowa

Spis treści

1 Wprowadzenie	2
1.1 Cel projektu	2
1.2 Wstępna wizja projektu	2
2 Metodologia wytwarzania	3
3 Analiza wymagań	4
3.1 Wymagania użytkownika i biznesowe	4
1. Klient:	4
2. Trener:	5
3. Portier:	5
4. Menedżer:	6
3.2 Wymagania funkcjonalne i нефункционалне	6
3.3 Przypadki użycia	6
1. Klient	6
2. Trener personalny	7
3. Portier	8
4. Menedżer	8
Dokładna specyfikacja kilku przypadków użycia	9
3.4 Potwierdzenie zgodności wymagań	13
4 Definicja architektury	13
5 Dane trwałe	16
5.1 Model logiczny danych	16
5.2 Przetwarzanie i przechowywanie danych	17
6 Specyfikacja analityczna i projektowa	18
Wykorzystane technologie:	18
Diagram klas	19
Diagram wdrożenia	20
7 Projekt standardu interfejsu użytkownika	21
Wykupienie terminu zajęć grupowych (use case Klient bb):	21
Wyświetlanie historii treningów (use case Klient g):	22
Wyszukanie klienta po ID klienta - numerze karty (use case Portier e)	22
8 Specyfikacja testów	23
9 Wirtualizacja/konteneryzacja	31
10 Bezpieczeństwo	32
11 Podręcznik użytkownika	32
12 Podręcznik administratora	34
13 Podsumowanie	36
14 Bibliografia	36

1 Wprowadzenie

1.1 Cel projektu

Celem projektu jest zaprojektowanie i stworzenie systemu zarządzania siecią siłowni. Tworzony przez nas system będzie posiadał czterech aktorów: klienta, trenera personalnego, portiera i menedżera. Każdy z nich będzie miał dostęp do swojego interfejsu aplikacji webowej z podstawowymi funkcjonalnościami niezbędnymi dla danej roli. Zadaniem aplikacji będzie usprawnienie działania siłowni poprzez automatyczne zarządzanie terminami, udostępnienie statystyk oraz umożliwienie użytkownikom łatwego dostępu do danych, których potrzebują do wykonywania swojej pracy.

1.2 Wstępna wizja projektu

System posiada czterech aktorów, których główne potrzeby i cele zdefiniowaliśmy następująco:

- Klient za pomocą naszego rozwiązania w łatwy sposób wybierze najbardziej odpowiednią do swoich potrzeb siłownię z sieci klubów fitness (na podstawie lokalizacji, usług - zajęć grupowych oferowanych w danym punkcie), odnajdzie i zapisze się na interesujące go zajęcia grupowe, a także sesje z wybranym przez siebie trenerem personalnym, dzięki czemu będzie w stanie realizować swoje cele treningowe. Aplikacja będzie również miejscem, gdzie w intuicyjny sposób dokona opłaty za wszelkiego rodzaju usługi, jakie są oferowane w naszej siłowni. Będzie miał w jednym miejscu dostęp do podglądu swojego terminarza na dany tydzień - w przejrzystej formie, co pozwoli mu na lepszą organizację czasu.

- Trener dzięki naszemu systemowi otrzyma platformę, na której będzie miał dostęp do wszystkich informacji potrzebnych mu do pracy. Nie tylko będzie miał wgląd do własnego grafiku pracy na dany tydzień, ale też otrzyma on dostęp do danych klientów, którzy wykupili z nim sesje treningowe. Informacje, takie jak aktualna waga, waga docelowa, preferowana liczba treningów w ciągu tygodnia oraz cel treningów pozwoli mu w sposób rzeczowy przygotować poszczególny plan ćwiczeń każdej sesji treningowej dla klienta.
- Portier naszej siłowni otrzyma narzędzie, dzięki któremu będzie w sposób jak najbardziej zautomatyzowany oraz efektywny weryfikował wejścia/wyjścia klientów. Dostępna dla niego jest intuicyjna lista klientów, która za pomocą albo numeru karty (równoważnego ID klienta) albo imienia i nazwiska umożliwi mu wyszukania klienta. Będzie mógł sprawdzić ważność jego karnetu - w razie potrzeby go aktywować czy odznaczyć wejścia klienta na siłownię. Wejścia i wyjścia na konkretne (przez nich wyspecyfikowane) zajęcia będą rejestrowane w systemie, tak samo jak i przydział/zwolnienie szafek przez odpowiednich klientów. W profilu klienta, do którego będzie miał on dostęp za pomocą wyżej wspomnianej listy, znajdzie on dane kontaktowe oraz tygodniowy grafik klienta
- Menedżer z kolei sprawuje kontrolę nad daną siłownią, więc ma łatwy dostęp do wszelkiego rodzaju statystyk, dzięki czemu może szybko reagować w przypadku jakichś nieprawidłowości.

2 Metodologia wytwarzania

Organizacja pracy w projekcie opiera się na regularnych spotkaniach (stacjonarnych), na których omawiane są efekty naszej wspólnej pracy, zaistniałe w procesie wytwarzania problemy, rozdzielane są bieżące zadania. Komunikacja odbywa się za pomocą Messengera (sprawy bieżące) oraz Teamsów (zdalne spotkania, konsultacje z ekspertami, spotkania z mentorem).

Wstępny podział zadań:

- Małgorzata Kozłowska - tworzenie frontendu trenera, portiera, menedżera, UX/UI
- Maria Polakowska - tworzenie backendu użytkownika, bazy danych
- Marta Sobol - tworzenie backendu trenera, portiera, menedżera, bazy danych
- Magda Twarowska - tworzenie frontendu użytkownika, UX/UI

Wiele rzeczy wykonujemy również wspólnie (na pierwszy etap – architekturę, funkcjonalności, dokumentację).

3 Analiza wymagań

3.1 Wymagania użytkownika i biznesowe

1. Klient:

Musi:

- mieć możliwość założenia konta w aplikacji webowej
- mieć możliwość podania swoich danych na swoim profilu użytkownika (imię, nazwisko, email, numer telefonu, płeć, wzrost, rok urodzenia, poziom zaawansowania, cel treningu, aktualna waga, waga docelowa, tygodniowa częstotliwość treningu {1-7 razy w tygodniu}, pożądany czas trwania jednego treningu {min. 30, max. 120 minut})
- mieć możliwość opłacenia jednorazowego wejścia na zajęcia, wykupienia karnetu czasowego (np.: 30 dni) z nieograniczoną liczbą wejść lub karnetu z liczbą wejść (np.: 20) nieograniczonego czasowo
- mieć możliwość zobaczenia aktualnego cennika karnetów uwzględniającego zniżki
- mieć możliwość zobaczenia całościowej oferty zajęć dostępnych w danym punkcie fitness (dostępne zajęcia grupowe, spotkanie z trenerem personalnym)
- mieć możliwość zobaczenia historii swoich ćwiczeń
- mieć możliwość zobaczenia informacji o wykonanym treningu (spalone kalorie, czas trwania)
- mieć możliwość wykupienia sesji treningowej z trenerem personalnym w wolnych terminach
- mieć możliwość odwołania zajęć do 24 godzin przed terminem ich rozpoczęcia
- mieć możliwość wykupienia zajęć grupowych w dostępnym terminie
- mieć możliwość zobaczenia listy klubów fitness w naszej sieci

Powinien:

- mieć możliwość zobaczenia zindywidualizowanego zestawu ćwiczeń stworzonego przez AI
- mieć możliwość zobaczenia historii swoich płatności
- mieć możliwość zmiany swoich danych
- mieć możliwość podania/modyfikacji swoich chorób w celu dostosowania do nich swoich planów treningowych
- mieć możliwość oceny pracy trenerów

Może:

- mieć możliwość zobaczenia informacji o wykonanym treningu (informacja zwrotna o jakości wykonania od AI na podstawie standardowych wartości bądź trenera, o ile został on wcześniej wykupiony)
- mieć możliwość zaplanowania treningu samodzielnie (wyświetlają się oszacowane statystyki - czas, liczba kcal)
- mieć możliwość dokonania zmian w zaplanowanym treningu
- mieć możliwość zobaczenia najbliższej siłowni / mapy siłowni
- mieć możliwość dodawania ćwiczeń do ulubionych

2. Trener:

Musi:

- mieć możliwość zobaczenia danych swoich podopiecznych którzy wykupili trening
- mieć możliwość zaplanowania/zmiany planów treningowych swoich podopiecznych
- mieć możliwość zobaczenia swoich godzin pracy tygodniowych
- mieć możliwość zobaczenia swoich zamówionych sesji (harmonogramu)

Powinien:

- mieć możliwość zobaczenia historii ćwiczeń swoich podopiecznych
- mieć możliwość dodawania istotnych uwag dla podopiecznych do wykonanych przez nich ćwiczeń
- mieć możliwość zobaczenia swojej pensji - pensji stałej oraz za zamówione sesje

Nie będzie:

- mieć możliwości zaplanowania zwolnienia

3. Portier:

Musi:

- mieć możliwość wyszukania klienta po numerze karty lub po nazwisku, i zobaczenia jego danych osobowych i danych jego karnetów
- mieć możliwość zarejestrowania wejścia/wyjścia klienta
- mieć możliwość aktywowania karnetu klienta przy pierwszym wejściu
- mieć możliwość przydzielenia wolnej szafki klientowi

Może:

- mieć możliwość doładowania konta użytkownika (użytkownik może zapłacić za usługi kartą/gotówką w portierni)

4. Menedżer:

Musi:

- mieć możliwość zobaczenia statystyk:
 - przedziały wiekowe klientów
 - częstość przychodzenia klientów (średnia liczba wejść na dany dzień oraz daną godzinę)
 - popularność karnetów (liczba zakupionych poszczególnych rodzajów karnetów),
 - popularność promocji (liczba zakupionych promocyjnych karnetów),
 - liczba zamówionych sesji u trenerów w stosunku do niezamówionych
 - częstość używania danego sprzętu (w godzinach)

Może:

- mieć możliwość dodawania nowych zniżek

3.2 Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne zostały zdefiniowane w punkcie 3.1. Poniżej znajdują się wymagania нефункционалне:

- aplikacja webowa
- dane przechowywane w relacyjnej bazie danych Oracle
- aplikacja skonteneryzowana
- system ma chronić wrażliwe dane użytkowników oraz hasła
- intuicyjny interfejs użytkowników
- monitorowanie aktywności użytkownika w czasie rzeczywistym (przesyłanie danych o wykonanych ćwiczeniach do bazy danych)

3.3 Przypadki użycia

1. Klient

- Rejestracja konta
- Logowanie do aplikacji
- Wylogowanie się
- Wyświetlanie swojego kalendarza treningów
- Wyświetlanie szczegółów swojej zaplanowanej sesji treningowej

- f. Odwoływanie zaplanowanych i opłaconych zajęć
- g. Wyświetlanie swojej historii treningów
- h. Wyświetlanie ćwiczeń i ich statystyk, które zostały wykonane w ramach jednego treningu
- i. Wyświetlanie swoich danych personalnych w profilu użytkownika
- j. Wyświetlanie swoich zakupionych karnetów
- k. Wyświetlanie szczegółów zakupionego karnetu
- l. Przeglądanie oferty dostępnych karnetów w sklepie
- m. Dodawanie karnetu do koszyka
- n. Usuwanie zakupionych, ale nieaktywowanych karnetów
- o. Wyświetlanie listy wszystkich siłowni

- p. Wyświetlanie oferty dostępnych zajęć grupowych
- q. Wyświetlanie opisów zajęć grupowych
- r. Wyświetlanie typów zajęć, które można kupić (grupowe / indywidualne)
- s. Wyświetlanie listy trenerów prowadzących zajęcia indywidualne w danej siłowni
- t. Wyświetlanie wolnych i kolizyjnych (dla klienta) terminów danych zajęć grupowych
- u. Wyświetlanie wolnych i kolizyjnych (dla klienta) terminów sesji indywidualnych u danego prowadzącego
- v. Wyświetlanie szczegółów danego terminu sesji indywidualnej
- w. Wyświetlanie szczegółów danego terminu zajęć grupowych
- x. Dodawanie danego terminu zajęć grupowych do koszyka
- y. Dodawanie danego terminu sesji indywidualnej do koszyka
- z. Wykupienie terminu sesji indywidualnej
- aa. Wykupienie terminu zajęć grupowych
- bb. Wyświetlanie elementów znajdujących się w koszyku
- cc. Wykupienie pozycji z koszyka
- dd. Usuwanie elementów z koszyka

2. Trener personalny

- a. Logowanie do aplikacji
- b. Wylogowanie się
- c. Wyświetlanie swojego kalendarza pracy - możliwych dostępnych terminów (na dany tydzień)

- d. Wyświetlanie swoich treningów (realizowanie w tym samym oknie co podpunkt c.)
- e. Wyświetlenie szczegółów na temat prowadzonych zajęć
- f. Wyświetlenie informacji klienta, który wykupił trening personalny
- g. Zaplanowanie następnego treningu (u osoby, która wykupiła trening personalny)
- h. Dodanie ćwiczenia do listy treningów danej sesji indywidualnej w oknie planowania treningu
- i. Usunięcie ćwiczenia z listy treningów w oknie planowania treningu
- j. Zmiana kolejności ćwiczenia w liście treningów - przesunięcie danego ćwiczenia do góry (będzie wykonywany wcześniej)
- k. Zmiana kolejności ćwiczenia w liście treningów - przesunięcie danego ćwiczenia do dołu (będzie wykonywany później)

3. Portier

- a. Logowanie do aplikacji
- b. Wylogowanie się
- c. Wyświetlanie bazy danych wszystkich klientów
- d. Wyszukanie klienta po imieniu i nazwisku
- e. Wyszukanie klienta po ID klienta - numerze karty
- f. Wyświetlanie danych kontaktowych klienta
- g. Wyświetlenie harmonogramu zajęć klienta
- h. Wyświetlenie karnetu klienta i danych z nim związanych (typ karnetu, status (aktywny/nieaktywny), data aktywacji/ważności, dostępna ilość wejść)
- i. Aktywowanie karnetu klienta
- j. Zarejestrowanie wejścia klienta na siłownię (wolny wstęp)
- k. Zarejestrowanie wejścia klienta na zajęcia grupowe/zajęcia z trenerem personalnym
- l. Przydzielenie szafki do klienta (bądź też nie)
- m. Zarejestrowanie wyjścia klienta z zajęć
- n. Zwolnienie szafki klienta

4. Menedżer

- a. Logowanie do aplikacji
- b. Wylogowanie się
- c. Wyświetlanie statystyki przedziałów wiekowych klientów
- d. Wyświetlanie statystyki częstości przychodzenia klientów

- e. Wyświetlanie statystyki popularności karnetów
- f. Wyświetlanie statystyki popularności promocji
- g. Wyświetlanie statystyki liczby zamówionych sesji u trenerów
- h. Wyświetlanie statystyki częstości używania danego sprzętu

Dokładna specyfikacja kilku przypadków użycia

USE CASE	Wyświetlanie historii treningów (use case Klient g)	
Description	Klient może zobaczyć wszystkie wykonane treningi posortowane datami	
Used by	Wyświetlanie ćwiczeń wykonanych w czasie wybranego treningu (use case Klient h)	
Preconditions	Klient jest zalogowany	
Success End Condition	Klient zapozna się z danymi odnośnie swoich treningów	
Failed End Condition	Klient nie będzie w stanie zapoznać się z danymi odnośnie swoich treningów	
Actors	Klient	
Trigger	-	
Basic Flow	Step	Action
	1	Logowanie do aplikacji (use case Klient b)
	2	W pasku bocznym wybiera "Statystyki"
	3	Wyświetla listę wszystkich swoich treningów z datami, czasami trwania i liczbą spalonych kalorii
Extensions	Step	Branching Action
	1a	Klient wpisał niepoprawne dane logowania 1a1. Podejmij kolejną próbę zalogowania (use case Klient b)

USE CASE	Wykupienie terminu zajęć grupowych (use case Klient bb)	
Description	Klient może kupić odpowiedni dla siebie termin zajęć grupowych	
Used by	-	
Preconditions	Klient jest zalogowany	
Success End Condition	Klient ma wykupione miejsce na zajęciach grupowych, siłownia otrzymała pieniądze, zamówione zajęcia są widoczne na stronie "Twoje zajęcia", dostępnej z paska bocznego	
Failed End Condition	Klient nie ma wykupionego miejsca na zajęciach grupowych, siłownia nie otrzymała pieniędzy	
Actors	Klient	
Trigger	-	
Basic Flow	Step	Action
	1	Logowanie do aplikacji (use case Klient b)
	2	Wyświetlanie typów zajęć, które można kupić - grupowe / indywidualne (use case Klient s) 1. Klika w pasku bocznym w 'Kup zajęcia'
	3	Klika w "Zajęcia grupowe"
	4	Wybiera siłownię, na której chce uczestniczyć w zajęciach
	5	Wyświetlanie wolnych i kolizyjnych (dla klienta) terminów danych zajęć grupowych (use case Klient u) 1. Wybiera typ zajęć ("Zumba", "Twist"...) 2. Sprawdza dostępne terminy
	6	Wyświetlanie szczegółów danego terminu sesji indywidualnej (use case Klient x) 1. Z widoku kalendarza wybiera dogodny dla siebie termin oraz prowadzącego (zaznaczone są kolizje z kalendarzem klienta)

		2. Sprawdza szczegóły zajęć
	7	Dodawanie danego terminu zajęć grupowych do koszyka (use case Klient y) 1. Dodaje termin do koszyka
	8	Wyświetla zawartość koszyka (use case Klient cc) 1. W pasku bocznym przechodzi do strony koszyka 2. Przegląda wszystkie zamówione pozycje i dostaje sumę do zapłaty
	9	Wykupuje pozycje z koszyka (use case Klient dd) 1. Dokonuje zakupu przez kliknięcie “Kupuję”
Extensions	Step	Branching Action
	6.1.a	Klient wybiera kolizyjny termin 6.2.a Na stronie szczegółów dostaje ostrzeżenie o kolizji oraz informacje o terminie, który chce kupić i terminie, z którym ma kolizję 7.1.a Pomimo kolizji dodaje zajęcia do koszyka 9.1.a Po kliknięciu ‘Kupuję’ dostaje kolejną informację o kolizji - tym razem musi podjąć decyzję, czy chce zamienić swoje zajęcia z kalendarza na nowe (te z koszyka), czy anuluje swój zakup 9.1.a.1 Jeśli anuluje zakup, wraca na stronę koszyka 9.1.a.2 Jeśli zdecyduje się na zmianę, transakcja dochodzi do skutku

USE CASE	Wyszukanie klienta po ID klienta - numerze karty (use case Portier e)	
Description	Portier może wyszukać danego klienta w bazie wszystkich klientów, korzystających z sieci siłowni po numerze karty.	
Used by	Wyświetlenie danych kontaktowych klienta (use case Portier f) Wyświetlenie harmonogramu zajęć klienta (use case Portier g)	
Preconditions	Portier jest zalogowany (use case Portier a)	
Success End Condition	Portier znajdzie po numerze karty klienta w bazie klientów siłowni	
Failed End Condition	Portier nie znajdzie po numerze karty klienta w bazie klientów siłowni	
Actors	Portier	
Trigger	-	
Basic Flow	Step	Action
	1	Logowanie do aplikacji (use case Portier a) 1. Klika "Zaloguj się"
	2	Wyświetlenie bazy danych wszystkich klientów (use case Portier c) 1. W widoku głównym klika na wyszukiwarkę
	3	Wyszukanie klienta po ID klienta - numerze karty (use case Portier e) 1. Wpisywanie kolejnych cyfr numeru karty (use case e.i.) (do końca numeru karty, bądź do momentu, w którym wyświetli się użytkownik - zależne od portiera)

Extensions	Step	Branching Action
	3a	<p>Portier wpisał niepoprawny numer karty klienta (nie istnieje taki klient w bazie danych)</p> <p>a.1 Portier klawiszem backspace usuwa niepoprawne cyfry</p> <p>a.2 Portier ponownie wpisuje cyfry - tym razem właściwą konfigurację (od końca poprawnego ciągu cyfr)</p> <p>lub</p> <p>b.1 Portier przyciskiem krzyżyka usuwa cały ciąg cyfr</p> <p>b.2 Portier zaczyna od początku wpisywać ciąg cyfr odpowiadający numerowi karty klienta</p>

3.4 Potwierdzenie zgodności wymagań

Zatwierdzam specyfikację wymagań, jako spełniających potrzeby Klienta.	<p>.....</p> <p>.....</p> <p>Data i podpis Właściciela tematu</p>
<i>Uwagi</i>	

4 Definicja architektury

W projekcie architektury zdecydowaliśmy się zastosować szablon architektury warstwowej.

Aplikacja składa się z trzech głównych modułów: backend, frontend oraz baza danych.

Implementacja symulowanej komunikacji ze sprzętami z siłowni oraz monitorowanie aktywności użytkownika zostało zaimplementowane za pomocą prostej aplikacji reactowej.

Backend

Stworzony za pomocą Pythonowego frameworka Django. Komunikuje się z frontendem za pomocą REST API (który używa protokołu HTTP) oraz z bazą danych przez Django ORM. Backend jest podzielony na 5 modułów: *client*, *trainer*, *manager*, *portier*, *simulation*. W każdym z tych modułów zawiera pliki *view* - służący do komunikacji z frontendem, *tworzy api*; *database* - służy do komunikacji z bazą; *tests* - testy dla danego modułu. Definicje modeli (odzwierciedlających tabele w bazie danych) znajdują się w oddzielnym katalogu *database_models*.

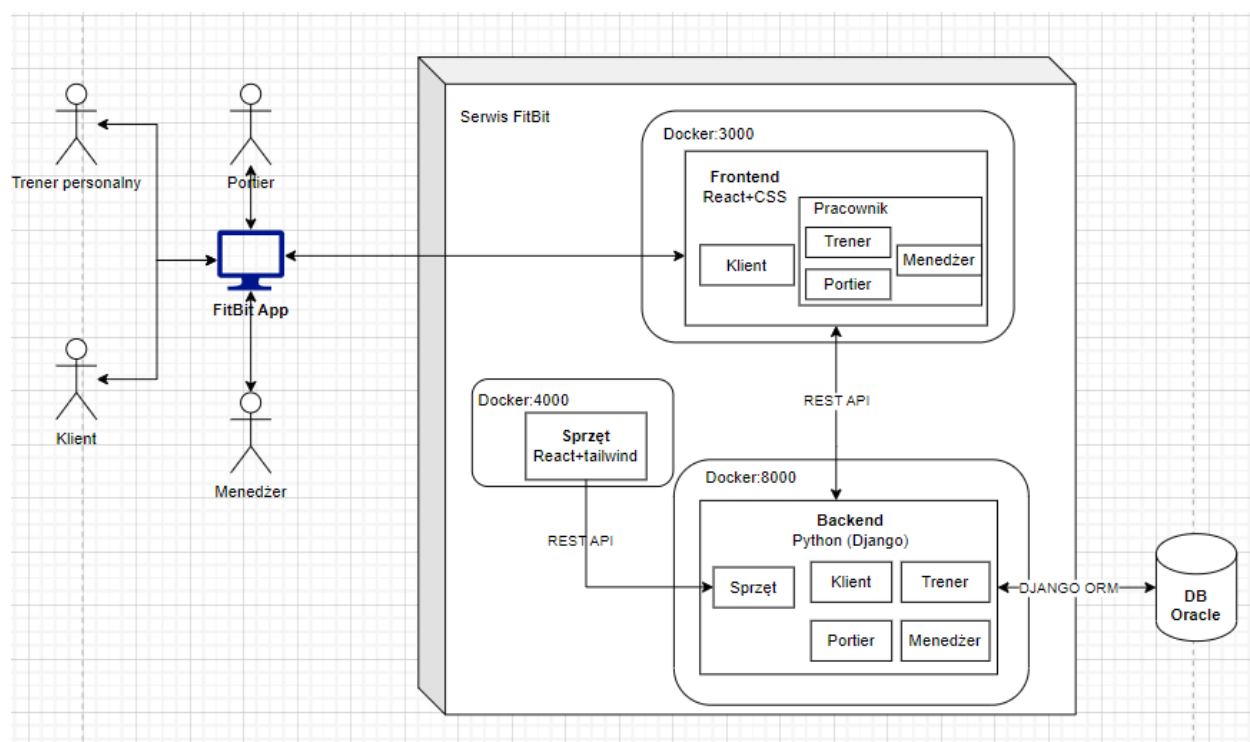
Baza danych

Relacyjna baza danych Oracle, z którą komunikujemy się za pomocą ORM Django.

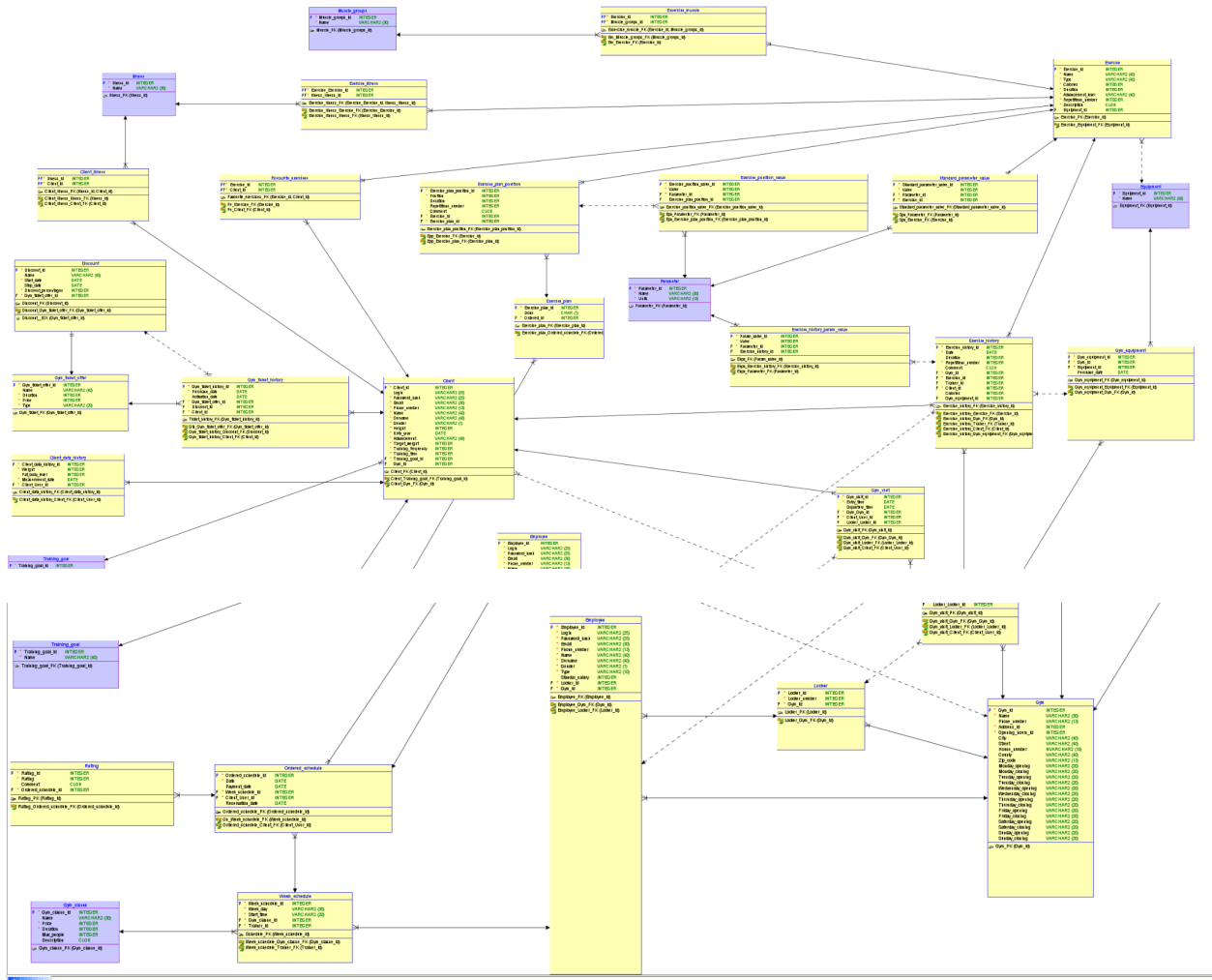
Frontend

Aplikacja webowa, stworzona za pomocą frameworku React, komunikuje się z backendem za pomocą Django REST API (który używa protokołu HTTP). Frontend podzieliliśmy na dwa główne moduły w katalogu *src* – *client* oraz *employee* (który dzieli się dalej na pracowników – *trainer*, *portier*, *manager*). Każdy z tych modułów zawiera podkatalogi: *components*, *functions*, *styles*, *views*, które zawierają potrzebne każdemu użytkownikowi komponenty, style oraz łączące je widoki. W katalogu *functions* znajdują się zapytania API.

Diagram modułów



Model relacyjny



Ze względu na rozmiary bazy zalecamy dokładniejszą analizę modelu, który znajduje się w repozytorium GitLab zespołu. Ścieżka do modelu: `sql/model/PZSP2.dmd`

5.2 Przetwarzanie i przechowywanie danych

W projekcie używamy bazy relacyjnej Oracle. Połączenie z bazą jest realizowane za pomocą ORM Django.

Większość funkcji i zapytań do bazy realizujemy w Django i to Django ORM nimi zarządza. W bazie dodaliśmy dwa triggerzy oraz 11 sekwencji. Sekwencje używamy, aby autoinkrementować klucze główne (id). Jeden trigger zarządza usuwaniem elementów z `ordered_schedule`, usuwa on powiązane wpisy w `exercise_plan` oraz `exercise_plan_position`. Drugi przy kupowaniu zajęć (dodawaniu wpisu do `ordered_schedule`), jeżeli dodawane zajęcia to trening indywidualny, dodaje powiązany wpis do `exercise_plan`.

6 Specyfikacja analityczna i projektowa

Link do repozytorium: https://gitlab.com/pzsp2_05/pzsp2_fitbit

Wykorzystane technologie:

- Języki programowania:
 - Python – zdecydowaliśmy się na ten język do implementacji backendu ze względu na jego prostotę, stabilność oraz łatwość tworzenia szeregu różnych wizualizacji, które w naszym projekcie odgrywają ważną rolę; dodatkowo ważnym aspektem jest obecność frameworku do tego języka – Django
 - CSS, JavaScript – do tworzenia frontendu postanowiliśmy użyć CSS oraz JS, ponieważ pozwalają na tworzenie intuicyjnych, interaktywnych i estetycznych stron internetowych; dodatkowo CSS jest kompatybilny z Figmą, co pozwoli nam przyspieszyć prace przy projektowaniu interfejsów, a JS posiada framework React
 - SQL
- Frameworki:
 - Django – do stworzenia backendu wybrałyśmy Django, ponieważ oferuje szeroki zestaw narzędzi, które ułatwiają tworzenie aplikacji webowych, m.in. ORM (praca z bazą danych), wbudowane bezpieczeństwo, Rest API (komunikacja z React)
 - React – do implementacji interfejsów użytkownika zdecydowaliśmy się na użycie frameworku JS – Reacta, ze względu na chęć jego lepszego poznania (co będzie możliwe w trakcie tworzenia projektu), jego łatwą integrację z Django i popularność wykorzystania na rynku w obecnych czasach
- Środowisko programowania: VSCode, SQL Developer
- Środowisko uruchamiania: Docker, localhost
- CI/CD:
 - GitLab CI/CD – jako narzędzie ciągłej integracji wybrałyśmy CI/CD oferowane przez GitLab, ponieważ tworzy zintegrowaną całość z innymi narzędziami GitLab, jest stosunkowo prosty w obsłudze oraz automatycznie wykrywa zmiany w repozytorium i uruchamia odpowiednie pipeline'y

Liczba testów jednostkowych: 215

Liczba plików (użyto narzędzia cloc)

Language	files	blank	comment	code
JSON	6	1	0	42344
XML	170	1	0	11225
Python	41	1366	1117	5848
JavaScript	84	684	46	4822
CSS	15	179	1	1097
SQL	3	103	42	511
YAML	2	16	0	63
Dockerfile	4	32	2	42
Markdown	2	33	0	40
HTML	1	3	20	20
SVG	1	0	0	1
SUM:	329	2418	1228	66013

Diagram klas

Z racji, że diagram jest dość duży – można się z nim wygodnie zapoznać w repozytorium, w katalogu głównym w pliku `my_models.png`

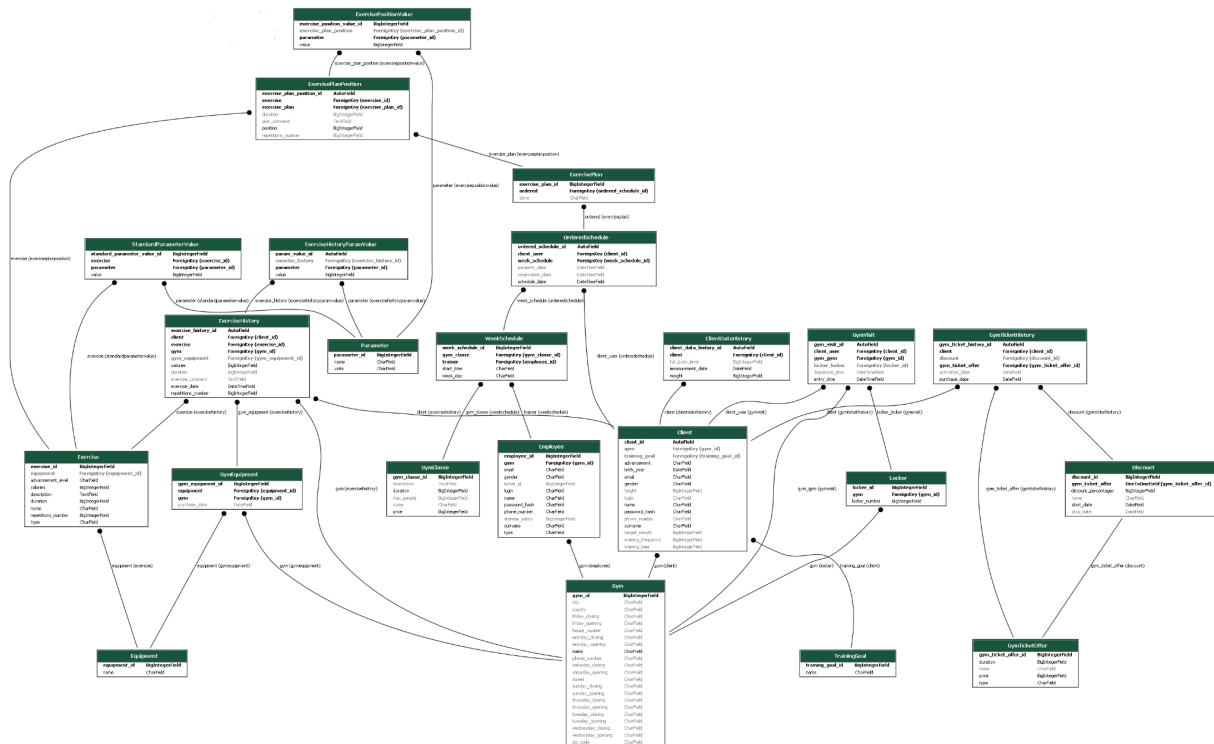
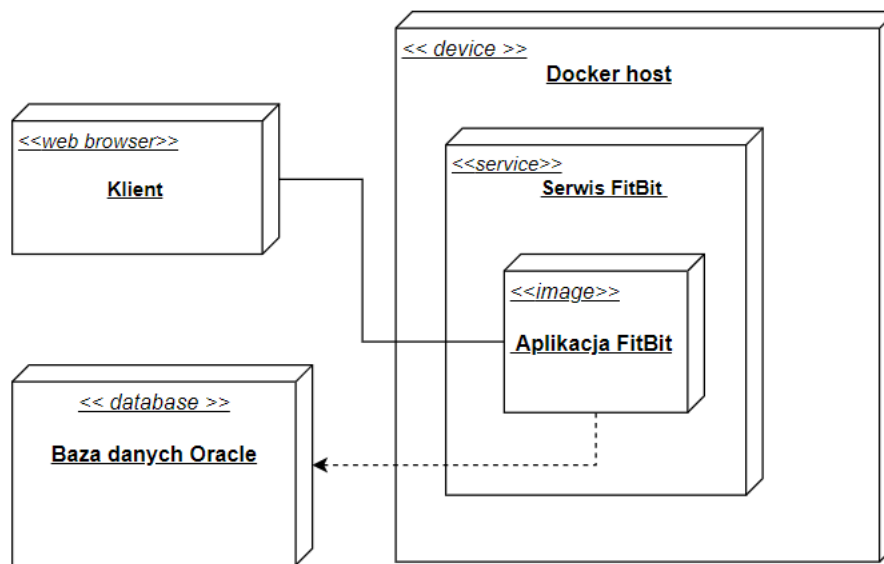


Diagram wdrożenia



7 Projekt standardu interfejsu użytkownika

Link do Figmy, w której znajduje się kompletny projekt interfejsów aktorów oraz przypadków użycia:

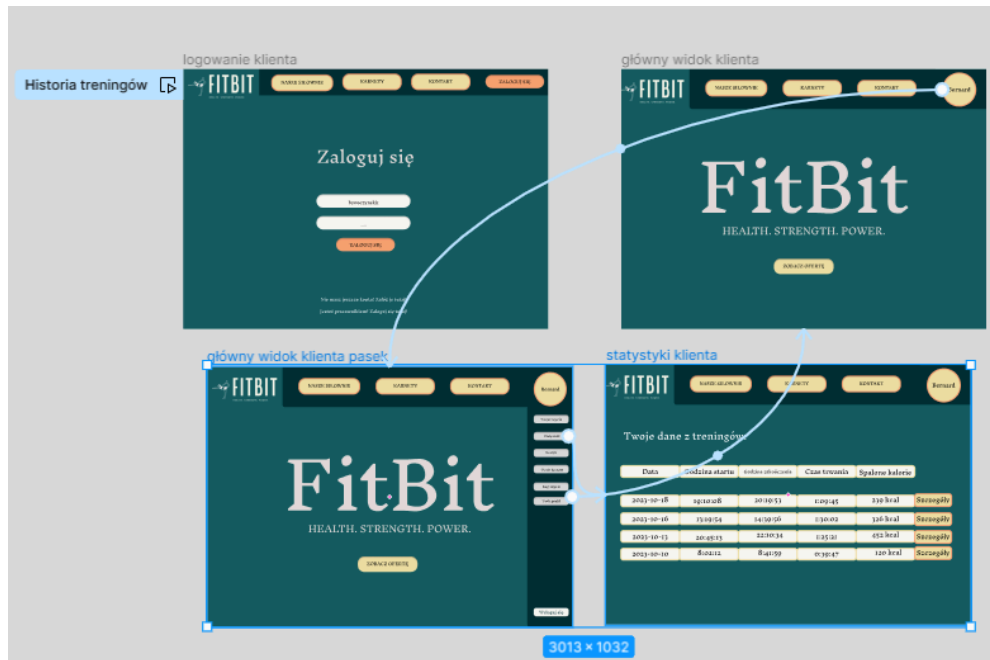
<https://www.figma.com/file/koTpS8LbudwuPe2H3w8O1O/Fitbit-app-prototype?type=design&node-id=1-13&mode=design&t=XYrxExAHYAb30GH1-0>

Poniżej kilka przykładowych przypadków użycia:

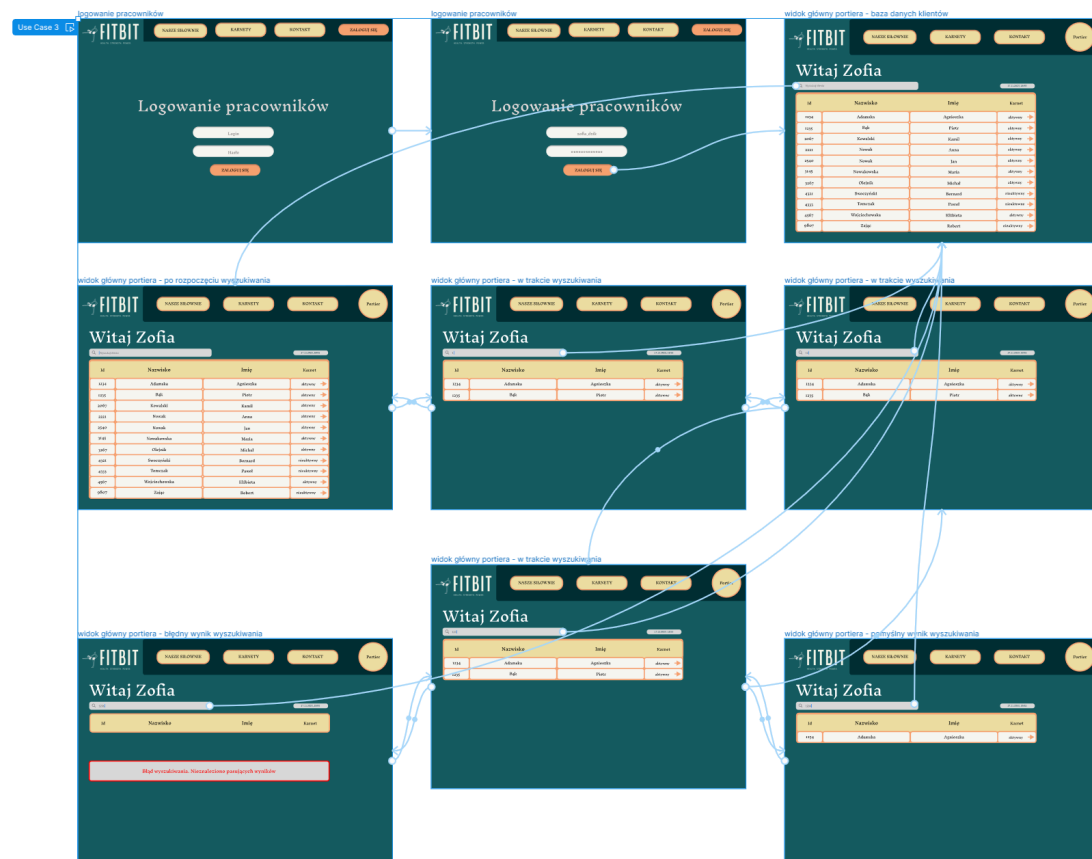
Wykupienie terminu zajęć grupowych (use case Klient bb):



Wyświetlanie historii treningów (use case Klient g):



Wyszukiwanie klienta po ID klienta - numerze karty (use case Portier e)



8 Specyfikacja testów

1. Testy jednostkowe

Testy jednostkowe są automatycznie przeprowadzane za pomocą narzędzia pipeline w repozytorium GitLab, sprawdzane jest tam również pokrycie linii kodu realizowane za pomocą narzędzia Coverage.py. Instrukcja uruchamiania testów lokalnie znajduje się w punkcie 12. w **Instrukcja uruchomienia testów jednostkowych**. Jako bazy testowej używamy SQLite. Nie używamy tej samej bazy do testów i aplikacji, ponieważ Django aby przeprowadzić testy tworzy nową bazę danych (aby podczas testów nie zepsuć tej produkcyjnej), a w udostępnionej przez Politechnikę bazie oracle nie mamy wystarczających uprawnień do tworzenia nowej bazy danych. Testy są napisane za pomocą TestCase oraz SimpleTestCase dostępnych w Django.

Pokrycie kodu

```
75 $ coverage report
76 Name                               Stmts  Miss  Cover
77 -----
78 client/admin.py                     1      0   100%
79 client/apps.py                      4      0   100%
80 client/client_error.py              6      0   100%
81 client/database.py                 358      5    99%
82 client/discount_calculate.py        7      0   100%
83 client/models.py                    1      0   100%
84 client/tests.py                     859      0   100%
85 client/urls.py                      3      0   100%
86 client/views.py                    265     35    87%
87 database_models/admin.py            1      0   100%
88 database_models/apps.py             4      0   100%
89 database_models/migrations/0001_initial.py  6      0   100%
90 database_models/models.py          220      0   100%
91 database_models/tests.py            1      0   100%
92 database_models/views.py            1      1     0%
93 django_backend/asgi.py              4      4     0%
94 django_backend/settings.py          29      0   100%
95 django_backend/settings_test.py     4      0   100%
96 django_backend/urls.py              4      0   100%
97 django_backend/wsgi.py              4      4     0%
98 manage.py                          12      2    83%
99 manager/admin.py                   1      0   100%
100 manager/apps.py                    4      0   100%
101 manager/database.py                271    195    28%
102 manager/models.py                   1      0   100%
103 manager/tests.py                   66      0   100%
104 manager/urls.py                     3      0   100%
105 manager/views.py                    57     33    42%
106 portier/admin.py                   1      0   100%
107 portier/apps.py                     4      0   100%
108 portier/database.py                115      0   100%
109 portier/models.py                   1      0   100%
110 portier/tests.py                   377      0   100%
111 portier/urls.py                     3      0   100%
112 portier/views.py                   61      0   100%
113 simulation/admin.py                 1      0   100%
114 simulation/apps.py                  4      0   100%
115 simulation/database.py              40      0   100%
116 simulation/models.py                1      0   100%
117 simulation/tests.py                 157      0   100%
118 simulation/urls.py                  3      0   100%
119 simulation/views.py                 41      0   100%
120 test.py                             50      0   100%
121 trainer/admin.py                    1      0   100%
122 trainer/apps.py                     4      0   100%
123 trainer/database.py                115      0   100%
124 trainer/models.py                   1      0   100%
125 trainer/tests.py                   350      0   100%
126 trainer/urls.py                     3      0   100%
127 trainer/views.py                    50      0   100%
128 views.py                           23      0   100%
129 -----
130 TOTAL                             3611     279    92%
```


Pokrycie kodu równe 92% oraz 215 testów jednostkowych, które wszystkie przechodzą.

2. Testy systemowe

Po ukończeniu prac nad aplikacją przyszła pora na testy systemowe. Realizowaliśmy je w następujący sposób:

1. Konfiguracja środowiska systemowego:
 - a. aplikację uruchomiłyśmy w Dockerze, żeby mieć niezależne środowisko, na które nie będzie wpływał system operacyjny
2. Scenariusze testowe klienta:
 - a. Rejestracja konta
 - i. klient uruchamia aplikację, klika w przycisk ‘ZALOGUJ’
 - ii. klika w link znajdujący się pod formularzem do logowania ‘*Nie masz jeszcze konta? Załóż je tutaj!*’
 - iii. w formularzu do rejestracji wypełnia wszystkie obowiązkowe dane (oznaczone gwiazdką) na stronie formularza dbając o ich poprawność (dane są na bieżąco walidowane)
 - iv. przechodzi do kolejnej strony formularza klikając na przycisk ‘PRZEJDŹ DALEJ’
 1. wyjątek: przy przechodzeniu z pierwszej strony formularza na drugą, walidowana jest unikalność loginu – jeśli w systemie dany login istnieje, wyświetla się adekwatny błąd
 - v. powtarza kroki *iii*, *iv*
 - vi. potwierdza poprzez kliknięcie ‘ZAREJESTRUJ SIĘ’
 - vii. konto zostało założone, następuje przeniesienie na stronę logowania
 - b. Logowanie do aplikacji
 - i. klient uruchamia aplikację, klika w przycisk ‘ZALOGUJ’
 - ii. wpisuje swoje dane logowania do formularza
 - iii. klika przycisk ‘ZALOGUJ’
 1. wyjątek: w przypadku niepoprawnych danych wyświetli się odpowiedni komunikat i klient nie będzie mógł się zalogować
 - iv. następuje przeniesienie na główną stronę aplikacji, klient jest zalogowany
 - c. Wylogowanie się
 - i. klient się loguje zgodnie ze scenariuszem **b**
 - ii. klient klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika ‘Wyloguj się’ w wyświetlonym sidebarze
 - iv. klient zostaje wylogowany, następuje przeniesienie na stronę główną

- d. Wyświetlanie swojego kalendarza treningów
 - i. klient się loguje zgodnie ze scenariuszem **b**
 - ii. klient klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika 'Twoje zajęcia'
 - iv. następuje przeniesienie do strony zakupionych zajęć klienta, które wyświetlane są tygodniowo (tygodnie można zmieniać przełącznikiem)
- e. Wyświetlanie szczegółów swojej zaplanowanej sesji treningowej
 - i. klient wykonuje kroki zgodnie ze scenariuszem **d**
 - ii. przy wybranych zajęciach klika przycisk 'Szczegóły'
 - iii. następuje przekierowanie na stronę szczegółów danych zajęć
- f. Odwoływanie zaplanowanych i opłaconych zajęć
 - i. klient wykonuje kroki zgodnie ze scenariuszem **e**
 - ii. klika znajdujący się pod tabelą ze szczegółami przycisk 'Anuluj zajęcia'
 - iii. zajęcia zostają anulowane, następuje przeniesienie na stronę 'Twoje karnety'
 - 1. wyjątek: klient chce odwołać zajęcia później, niż 24h przed ich rozpoczęciem – jest to niedozwolone, więc wyświetla się adekwatna informacja, a przycisk anulowana zostaje zablokowany
- g. Wyświetlanie swojej historii treningów
 - i. klient się loguje zgodnie ze scenariuszem **b**
 - ii. klient klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika 'Statystyki'
 - iv. następuje przeniesienie do strony informującej o wejściach i odbytych na siłowni treningach klienta
- h. Wyświetlanie ćwiczeń i ich statystyk, które zostały wykonane w ramach jednego treningu
 - i. klient wykonuje kroki zgodnie ze scenariuszem **g**
 - ii. przy wybranym treningu klika przycisk 'Szczegóły'
 - iii. następuje przeniesienie na stronę szczegółów, gdzie klient może sprawdzić wykonane w danej sesji ćwiczenia oraz ich statystyki
- i. Wyświetlanie swoich danych personalnych w profilu użytkownika
 - i. klient się loguje zgodnie ze scenariuszem **b**
 - ii. klient klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika 'Profil klienta'

- iv. następuje przeniesienie do strony profilu klienta, gdzie może zobaczyć wszystkie podane przez siebie podczas rejestracji dane
- j. Wyświetlanie swoich zakupionych karnetów
 - i. klient się loguje zgodnie ze scenariuszem **b**
 - ii. klient klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika 'Twoje karnety'
 - iv. następuje przeniesienie do strony wyświetlającej wszystkie zakupione przez klienta karnety oraz ich status
- k. Wyświetlanie szczegółów zakupionego karnetu
 - i. klient wykonuje kroki zgodnie ze scenariuszem **j**
 - ii. przy wybranym karnecie klika przycisk 'Szczegóły'
 - iii. następuje przeniesienie na stronę szczegółów, gdzie klient może sprawdzić dokładne daty ważności karnetu, jego status oraz ceny
- l. Usuwanie zakupionych, ale nieaktywowanych karnetów
 - i. klient wykonuje kroki zgodnie ze scenariuszem **k** dla dowolnego, nieaktywowanego karnetu
 - ii. klika w przycisk 'Anuluj zakup karnetu'
 - iii. karnet zostaje usunięty z zakupionych karnetów klienta, następuje przekierowanie do strony wszystkich zakupionych karnetów
- m. Przeglądanie oferty dostępnych karnetów w sklepie
 - i. klient klika w headerze w przycisk 'KARNETY'
 - ii. następuje przekierowanie do sklepu karnetów
- n. Dodawanie karnetu do koszyka
 - i. klient loguje się zgodnie ze scenariuszem **b**
 - ii. klient klika w headerze w przycisk 'KARNETY'
 - iii. następuje przekierowanie do sklepu karnetów
 - iv. klika przycisk koszyka obok wybranego karnetu
 - v. karnet zostaje dodany do koszyka, wyświetlony zostaje komunikat
- o. Wyświetlanie listy wszystkich siłowni
 - i. klient klika w headerze w przycisk 'NASZE SIŁOWNIE'
 - ii. następuje przekierowanie do listy siłowni znajdujących się w naszej sieci
- p. Wyświetlanie listy trenerów prowadzących zajęcia indywidualne w danej siłowni
 - i. klient postępuje zgodnie ze scenariuszem **o**
 - ii. klika w wybraną pozycję w liście
 - iii. wyświetleni zostają trenerzy prowadzący zajęcia na danej siłowni

- q. Wyświetlanie oferty dostępnych zajęć na danej siłowni
 - i. klient postępuje zgodnie ze scenariuszem **o**
 - ii. klika w wybraną pozycję w liście
 - iii. wyświetlone zostają zajęcia odbywające się w danej siłowni
- r. Wyświetlanie typów zajęć, które można kupić (grupowe / indywidualne)
 - i. klient się loguje zgodnie ze scenariuszem **b**
 - ii. klient klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika 'Kup zajęcia'
 - iv. następuje przekierowanie na stronę decyzyjną, gdzie wyświetlone są dostępne typy zajęć
- s. Wyświetlanie wolnych i kolizyjnych (dla klienta) terminów danych zajęć grupowych
 - i. klient postępuje zgodnie ze scenariuszem **r**
 - ii. klika 'Zajęcia grupowe'
 - iii. wybiera dowolną siłownię
 - iv. wybiera z puli dostępnych na wybranej wyżej siłowni zajęcia takie, na które chce iść
 - v. wyświetlają się wszystkie zajęcia grupowe dostępne w danym tygodniu w danej siłowni; tygodnie można zmieniać przełącznikiem; czerwonym kolorem oznaczono terminy, na których nie ma już miejsc, bądź z którymi klient ma kolizje
- t. Wyświetlanie wolnych i kolizyjnych (dla klienta) terminów sesji indywidualnych u danego prowadzącego
 - i. klient postępuje zgodnie ze scenariuszem **r**
 - ii. klika 'Zajęcia indywidualne'
 - iii. wybiera dowolną siłownię
 - iv. wybiera z puli dostępnych na wybranej wyżej siłowni trenerów takiego, z którym chce realizować trening
 - v. wyświetlają się wszystkie zajęcia indywidualne dostępne w danym tygodniu w danej siłowni prowadzone przez danego trenera; tygodnie można zmieniać przełącznikiem; czerwonym kolorem oznaczono terminy, na których nie ma już miejsc, bądź z którymi klient ma kolizje
- u. Wyświetlanie szczegółów danego terminu sesji indywidualnej
 - i. klient postępuje zgodnie ze scenariuszem **t**
 - ii. przy wybranym terminie zajęć klika 'Szczegóły'

- iii. następuje przekierowanie na stronę szczegółów, gdzie klient widzi zarówno szczegóły wybranego terminu, jak i szczegóły terminu z którym ten ewentualnie ma kolizję (z zakupionych zajęć bądź z koszyka; jeżeli nie ma miejsc, jest wyświetlony odpowiedni komunikat)
- v. Wyświetlanie szczegółów danego terminu zajęć grupowych
 - i. klient postępuje zgodnie ze scenariuszem **s**
 - ii. przy wybranym terminie zajęć klika 'Szczegóły'
 - iii. następuje przekierowanie na stronę szczegółów, gdzie klient widzi zarówno szczegóły wybranego terminu, jak i szczegóły terminu z którym ten ewentualnie ma kolizję (z zakupionych zajęć bądź z koszyka; jeżeli nie ma miejsc, jest wyświetlony odpowiedni komunikat)
- w. Dodawanie danego terminu zajęć grupowych do koszyka
 - i. klient postępuje zgodnie ze scenariuszem **v**
 - ii. klika przycisk 'Dodaj do koszyka'
 - iii. zajęcia zostają dodane do koszyka, zostaje wyświetlony komunikat, przycisk 'Dodaj do koszyka' zostaje zablokowany
 - 1. wyjątek: w przypadku kolizji, braku miejsc bądź gdy zajęcia są już w koszyku, następuje wyświetlenie odpowiedniego komunikatu
- x. Dodawanie danego terminu sesji indywidualnej do koszyka
 - i. klient postępuje zgodnie ze scenariuszem **u**
 - ii. klika przycisk 'Dodaj do koszyka'
 - iii. zajęcia zostają dodane do koszyka, zostaje wyświetlony komunikat, przycisk 'Dodaj do koszyka' zostaje zablokowany
 - 1. wyjątek: w przypadku kolizji, braku miejsc bądź gdy zajęcia są już w koszyku, następuje wyświetlenie odpowiedniego komunikatu
- y. Wyświetlanie elementów znajdujących się w koszyku
 - i. jeśli nie jest jeszcze zalogowany, to klient się loguje zgodnie ze scenariuszem **b**
 - ii. klient klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika 'Koszyk'
 - iv. następuje przekierowanie do strony koszyka, gdzie są wyświetlone wszystkie dodane tam zajęcia oraz karnety
- z. Wykupienie terminu sesji indywidualnej

- i. klient postępuje zgodnie ze scenariuszem x, a następnie y
 - ii. klika przycisk ‘Wykup koszyk’
 - iii. następuje przekierowanie na stronę rezerwacji zamówienia, gdzie klient ma 10 minut na wypełnienie danych do przelewu i zatwierdzenie
 - 1. wyjątek: na zajęciach nie ma już miejsc – zostanie wyświetlony odpowiedni komunikat, a klient nie będzie mógł wykupić koszyka, dopóki własnoręcznie nie usunie tamtych zajęć
- aa. Wykupienie terminu zajęć grupowych
 - i. klient postępuje zgodnie ze scenariuszem w, a następnie y
 - ii. klika przycisk ‘Wykup koszyk’
 - iii. następuje przekierowanie na stronę rezerwacji zamówienia, gdzie klient ma 10 minut na wypełnienie danych do przelewu i zatwierdzenie
 - 1. wyjątek: na zajęciach nie ma już miejsc – zostanie wyświetlony odpowiedni komunikat, a klient nie będzie mógł wykupić koszyka, dopóki własnoręcznie nie usunie tamtych zajęć
- bb. Usuwanie elementów z koszyka
 - i. klient postępuje zgodnie ze scenariuszem y
 - ii. obok wybranej pozycji klika ‘Usuń’
 - iii. pozycja zostaje usunięta
- cc. Wyświetlenie cennika zajęć
 - i. klient klika w headerze ‘CENNIK’
 - ii. następuje przekierowanie na stronę cennika, gdzie wyświetlone są wszystkie zajęcia oraz ich ceny
- dd. Wyświetlenie szczegółów zajęć dodanych do koszyka
 - i. klient postępuje zgodnie ze scenariuszem y
 - ii. obok wybranej pozycji klika ‘Szczegóły’
 - iii. następuje przekierowanie na stronę szczegółów danych zajęć
- 3. Scenariusze testowe trenera
 - a. Logowanie do aplikacji
 - i. trener uruchamia aplikację, klika w przycisk ‘ZALOGUJ’
 - ii. następnie klika w przycisk ‘Jesteś pracownikiem? Zaloguj się tutaj!’, znajdujący się poniżej głównego panelu logowania
 - iii. wpisuje swoje dane logowania do formularza
 - iv. klika przycisk ‘ZALOGUJ SIĘ’

1. wyjątek: w przypadku niepoprawnych danych wyświetli się komunikat “Błędne dane logowania ”i trener nie będzie mógł się zalogować
- v. następuje przeniesienie na główną stronę aplikacji, trener jest zalogowany
- b. Wylogowanie się
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. trener klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika ‘Wyloguj się’ w wyświetlonym sidebarze
 - iv. trener zostaje wylogowany, następuje przeniesienie na stronę główną
- c. Wyświetlenie tygodniowego harmonogramu zajęć
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie startowej wyświetli się kalendarz przedstawiający harmonogram wszystkich zajęć prowadzonych przez trenera w danym tygodniu
- d. Wyświetlenie agendy na dany dzień
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie startowej wyświetli się kalendarz przedstawiający harmonogram wszystkich zajęć prowadzonych przez trenera
 - iii. w górnym rzędzie kalendarza, po prawej stronie znajduje się przycisk Agenda - klika w niego
 - iv. wyświetli się lista zajęć na dany dzień
- e. Wyświetlenie harmonogramu na dany dzień
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie startowej wyświetli się kalendarz przedstawiający harmonogram wszystkich zajęć prowadzonych przez trenera
 - iii. w górnym rzędzie kalendarza, po prawej stronie znajduje się przycisk Day - klika w niego
 - iv. Wyświetla się harmonogram na dany dzień (w postaci kalendarza)
- f. Wyświetlenie szczegółów prowadzonych przez siebie zajęć grupowych
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie startowej wyświetli się kalendarz przedstawiający harmonogram wszystkich zajęć prowadzonych przez trenera
 - iii. trener wybiera z kalendarza zajęcia grupowe
 - iv. klika w “Zobacz szczegóły zajęć grupowych: <Nazwa zajęć grupowych>”
 - v. trenerowi wyświetla się strona zajęć grupowych z szczegółowym ich opisem

- g. Wyświetlenie szczegółów klientów, z którymi prowadzi treningi indywidualne
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie startowej wyświetli się kalendarz przedstawiający harmonogram wszystkich zajęć prowadzonych przez trenera
 - iii. trener wybiera z kalendarza zajęcia indywidualne
 - iv. klika w “Zobacz szczegóły zajęć indywidualnych z: <imię klienta>”
 - 1. Wyjątek: jeśli zajęcia indywidualne nie zostały wykupione to żadne imię klienta się nie wyświetli (w nazwie przycisku) - w panelu klienta będą puste pola
 - v. Wyświetli się panel z informacjami o kliencie
 - h. Zaplanowanie następnego treningu
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie startowej wyświetli się kalendarz przedstawiający harmonogram wszystkich zajęć prowadzonych przez trenera
 - iii. trener wybiera z kalendarza zajęcia indywidualne
 - iv. klika w “Zobacz szczegóły zajęć indywidualnych z: <imię klienta>”
 - v. Klika w “Zaplanuj następny trening”
 - vi. Po wybraniu odpowiednich ćwiczeń, ich powtórzeń oraz czasu oraz ich dodaniu, trener może zapisać trening - klika w “Zapisz trening!”
 - vii. Trening został zapisany
 - i. Modyfikowanie następnego treningu
 - i. trener loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie startowej wyświetli się kalendarz przedstawiający harmonogram wszystkich zajęć prowadzonych przez trenera
 - iii. trener wybiera z kalendarza zajęcia indywidualne
 - iv. klika w “Zobacz szczegóły zajęć indywidualnych z: <imię klienta>”
 - v. Klika w “Zaplanuj następny trening”
 - vi. Trener może zmienić kolejność ćwiczeń za pomocą strzałek, zmienić liczbę powtórzeń, czas oraz ćwiczenia. Może także usunąć wybrane przez siebie z listy.
 - vii. Klika w “Zapisz trening!”
 - viii. Modyfikacja treningu została zapisana.
4. Scenariusze testowe portiera
- a. Logowanie do aplikacji
 - i. portier uruchamia aplikację, klika w przycisk ‘ZALOGUJ’

- ii. następnie klika w przycisk ‘Jesteś pracownikiem? Zaloguj się tutaj!’, znajdujący się poniżej głównego panelu logowania
- iii. wpisuje swoje dane logowania do formularza
- iv. klika przycisk ‘ZALOGUJ SIĘ’
 - 1. wyjątek: w przypadku niepoprawnych danych wyświetli się komunikat “Błędne dane logowania” i portier nie będzie mógł się zalogować
- v. następuje przeniesienie na główną stronę aplikacji, portier jest zalogowany
- b. Wylogowanie się
 - i. portier loguje się zgodnie ze scenariuszem **a**
 - ii. portier klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika ‘Wyloguj się’ w wyświetlonym sidebarze
 - iv. portier zostaje wylogowany, następuje przeniesienie na stronę główną
- c. Wyszukanie klienta z listy
 - i. portier loguje się zgodnie ze scenariuszem **a**
 - ii. na stronie głównej, w wyszukiwarce wpisuje klucz, po którym chce wyszukać klienta (imię, nazwisko, id, numer telefonu)
 - iii. wyświetli się szukany klient
 - 1. wyjątek: w przypadku niepoprawnych danych (klucza) zwrócona lista będzie pusta
- d. Wyświetlenie informacji klienta
 - i. portier loguje się zgodnie ze scenariuszem **a**
 - ii. po znalezieniu klienta (scenariusz **c**) klika w strzałkę w kolumnie “Zobacz więcej informacji”
 - iii. Wyświetli się panel informacyjny
- e. Aktywacja karnetu
 - i. portier loguje się zgodnie ze scenariuszem **a**
 - ii. po znalezieniu klienta (scenariusz **c**) klika w strzałkę w kolumnie “Zobacz więcej informacji”
 - iii. Wyświetli się panel informacyjny
 - iv. W Dane karnetu portier klika w rozwijaną listę dostępnych karnetów, jakie wykupił klient
 - 1. Wyjątek: w przypadku, gdy klient nie wykupił żadnego karnetu, lista rozwijana będzie pusta
 - v. Po wybraniu karnetu klika w “Aktywuj karnet:<nazwa karnetu>”!
 - vi. Karnet zostaje aktywowany

- f. Rejestracja wejścia
 - i. portier loguje się zgodnie ze scenariuszem **a**
 - ii. po znalezieniu klienta (scenariusz **c**) klika w strzałkę w kolumnie “Zobacz więcej informacji”
 - iii. Wyświetli się panel informacyjny
 - iv. Klika w przycisk: ‘Rozpocznij trening’
 - v. Zarejestrowano wejście klienta
 - g. Przydzielenie szafki
 - i. portier loguje się zgodnie ze scenariuszem **a**
 - ii. po znalezieniu klienta (scenariusz **c**) klika w strzałkę w kolumnie “Zobacz więcej informacji”
 - iii. Wyświetli się panel informacyjny
 - iv. Klika w przycisk: ‘Przydziel szafkę’
 - 1. Wyjątek: w przypadku nie zarejestrowania wcześniej wejścia, wyświetli się odpowiedni komunikat mówiący, że nie jest możliwe przydzielenie szafki bez uprzednio zarejestrowania wejścia klienta
 - v. Portier przydzielił szafkę klientowi
 - h. Rejestracja wyjścia
 - i. portier loguje się zgodnie ze scenariuszem **a**
 - ii. po znalezieniu klienta (scenariusz **c**) klika w strzałkę w kolumnie “Zobacz więcej informacji”
 - iii. Wyświetli się panel informacyjny
 - iv. Klika w przycisk ‘Zakończ trening’
 - v. Poprawnie zarejestrowano wyjście klienta
5. Scenariusze testowe menedżera
- a. Logowanie do aplikacji
 - i. menedżer uruchamia aplikację, klika w przycisk ‘ZALOGUJ’
 - ii. następnie klika w przycisk ‘Jesteś pracownikiem? Zaloguj się tutaj!’, znajdujący się poniżej głównego panelu logowania
 - iii. wpisuje swoje dane logowania do formularza
 - iv. klika przycisk ‘ZALOGUJ SIĘ’
 - 1. wyjątek: w przypadku niepoprawnych danych wyświetli się komunikat “Błędne dane logowania” i menedżer nie będzie mógł się zalogować
 - v. następuje przeniesienie na główną stronę aplikacji, menedżer jest zalogowany
 - b. Wylogowanie się

- i. menedżer loguje się zgodnie ze scenariuszem **a**
 - ii. menedżer klika przycisk swojego profilu znajdujący się w headerze
 - iii. klika ‘Wyloguj się’ w wyświetlonym sidebarze
 - iv. menedżer zostaje wylogowany, następuje przeniesienie na stronę główną
- c. Wyświetlenie odpowiedniej statystyki (dostępnej z listy)
 - i. menedżer loguje się zgodnie ze scenariuszem **a**
 - ii. z lewego panelu zawierającego listę dostępnych statystyk (tytuł: “Dostępne statystyki”) może wybrać tę, która go interesuje - po kliknięciu w nazwę po chwili, po prawej stronie, wyświetli się statystyka zawierająca najnowsze dane sieci klubów fitness, przedstawione za pomocą grafów.
 1. W przypadku wybrania statystyki obciążenie sprzętu przed wyświetleniem statystyki menedżer musi wybrać interesujący go sprzęt z listy dostępnych (Wybierz sprzęt).

Wszystkie powyższe scenariusze zostały przez nas przeprowadzone i zakończyły się pozytywnym rezultatem.

9 Wirtualizacja/konteneryzacja

Aplikacja została skonteneryzowana za pomocą Dockera. Wydzieliliśmy trzy oddzielne kontenery: dla frontendu, backendu oraz aplikacji do symulacji sprzętu. Ponadto, stworzyliśmy plik docker-compose, aby w łatwy i szybki sposób móc budować i uruchamiać te trzy kontenery.

Użyte obrazy:

- frontend: node:20-alpine
- backend: python:3.8.13-bullseye
- aplikacja do symulacji: node:20-alpine

Ponadto, specjalnie do testów stworzyliśmy oddzielny kontener (obraz python:3.8.13-bullseye) z odpowiednią konfiguracją potrzebną do testów. Nie został on dołączony do pliku docker-compose, aby nie uruchamiać niepotrzebnie testów przy każdym uruchomieniu aplikacji.

10 Bezpieczeństwo

Użytkownika będziemy autoryzować za pomocą loginu i hasła. Będziemy korzystać z mechanizmów bezpieczeństwa zapewnionych w Django. Framework dostarcza szereg

bezpiecznych funkcji do manipulowania danymi w bazie, które są zoptymalizowane pod kątem bezpieczeństwa i zapobiegają np. atakom SQL injection. Dzięki parametryzacji zapytań eliminujemy możliwość wstrzykiwania kodu, co jest o wiele bezpieczniejsze od bezpośredniego wstawiania danych do zapytania SQL. Ponadto Django automatycznie waliduje wprowadzane dane przed zapisaniem ich do bazy danych. W bazie będziemy przechowywać hash danych wrażliwych zamiast ich surowej wersji.

Do połączenia z bazą oraz wysyłania email potrzebne jest podanie hasła. Nasze repozytorium jest publiczne i wielce niewskazane byłoby przechowywanie tam pliku z hasłami. Z tego powodu, przed uruchomieniem aplikacji należy stworzyć plik .env zawierający wszystkie potrzebne hasła. Jest to bezpieczniejsze, a hasła będą przekazywane bezpośrednio klientom, a nie dostępne publicznie w repozytorium.

11 Podręcznik użytkownika

1. Aby użytkownik mógł korzystać z aplikacji powinien posiadać odpowiednie konto (klienckie lub pracownicze); nowy klient siłowni może w każdej chwili zarejestrować się w systemie.
2. Bez zalogowania się użytkownik ma dostęp do następujących stron (dostępne poprzez przyciski w headerze): lista siłowni, cennik karnetów oraz cennik zajęć.
 - a. *Siłownie*: wyświetla nazwy i adresy siłowni, które funkcjonują w sieci; po kliknięciu na dany rząd wyświetla pracowników klikniętego obiektu oraz zajęcia, które są tam prowadzone
 - b. *Karnety*: wyświetla wszystkie karnety obecne w ofercie oraz ich ceny; po kliknięciu koszyka będąc niezalogowanym, strona przekieruje na ekran logowania
 - c. *cennik*: wyświetla cennik zajęć (każdy typ zajęć na każdej siłowni ma taką samą cenę)
3. **Klient**: Po poprawnym zalogowaniu, centrum komunikacji aplikacji znajduje się w pasku bocznym – użytkownik może za jego pomocą przekierować się na następujące strony:
 1. *Twoje zajęcia*: klient może wyświetlić informacje dot. swoich zakupionych zajęć oraz wyświetlić ich szczegóły
 2. *Statystyki*: klient ma dostęp do danych ze wszystkich swoich treningów na siłowni; po wejściu w szczegóły widzi wszystkie ćwiczenia wykonane w ramach danej sesji oraz ich statystyki
 3. *Koszyk*: klient może wyświetlić wszystkie zajęcia oraz karnety, które dodał do koszyka oraz je wykupić; przy wykupieniu może się okazać, że po dodaniu zajęć do koszyka zostały w międzyczasie wykupione wszystkie

- miejsca – wtedy zwrócona zostanie informacja o braku miejsc i klient zostanie poproszony o usunięcie tych zajęć z koszyka
4. *Twoje karnety*: klient ma dostęp do wszystkich swoich wykupionych kiedykolwiek karnetów, widzi ich obecny status (aktywny, wygasły, nieaktywowany) oraz inne informacje
 5. *Kup zajęcia*: klient może wykupić dowolne zajęcia, w których ma ochotę uczestniczyć; na początku wybiera typ zajęć (indywidualne/grupowe), a następnie po przekierowaniu ustawia preferowaną siłownię oraz prowadzącego (w przypadku zajęć indywidualnych); następnie wyświetla się lista zajęć (brak miejsc oraz kolizje z zakupionymi zajęciami klienta oznaczone kolorem czerwonym) – klient może wejść w szczegóły danego terminu, gdzie będzie możliwość dodania go do koszyka, informacja o kolizji z koszykiem/zajęciami klienta lub braku miejsc
 6. *Profil klienta*: klient może wyświetlić informacje, które podał przy rejestracji
 7. *Wyloguj*: klient zostanie wylogowany (uwaga! – przy wylogowaniu zawartość koszyka jest usuwana, więc po ponownym zalogowaniu na to samo konto koszyk będzie pusty)

Zalogowany użytkownik ma też oczywiście dostęp do stron z headera: siłowni, sklepu karnetów, gdzie może dodać do koszyka karnet dla siebie, oraz cennika zajęć.

4. Trener:

- a. Po poprawnym zalogowaniu na *stronie głównej* trenera wyświetli kalendarz z tygodniowym harmonogramem zajęć.
- b. Po wybraniu zajęć grupowych, wyświetli się przycisk *Zobacz szczegóły zajęć grupowych*. Po kliknięciu wyświetli się strona zajęć, z ich nazwą i opisem.
- c. Po wybraniu treningu indywidualnego wyświetli się informacja, z kim te zajęcia się odbywają (o ile dany termin został wykupiony) - trener wówczas może przejść w *Zobacz szczegóły zajęć indywidualnych z: <imię klienta>*.
- d. Po wejściu w *Zobacz szczegóły zajęć indywidualnych z: <imię klienta>* wyświetlą się *Dane klienta*, *Cele treningowe* oraz *Preferencje treningowe*.
- e. Ze strony szczegółów klienta można przejść do *Zaplanuj następny trening* - otworzy się wówczas widok *Zaplanuj trening*, gdzie można przygotować plan treningowy dla danego klienta, wybierając dane ćwiczenie z listy ćwiczeń, wraz z odpowiednią jego liczbą powtórzeń/czasem. Można tak zdefiniowane ćwiczenie dodać, edytować, usunąć oraz przesunąć względem innych ćwiczeń w liście. Przygotowany trening można zapisać - wówczas będzie on widoczny, gdy ponownie wejdziemy w informacje szczegółowe klienta oraz w zakładkę *Zaplanuj nadchodzący trening*

- f. Na koniec z panelu bocznego trener może się wylogować, wciskając *Wyloguj się*.

5. **Portier:**

- a. Po poprawnym zalogowaniu na *stronie głównej* portiera wyświetli lista klientów - wyświetlą się informacje takie jak Id, Numer telefonu, Nazwisko, Imię, Status karty oraz *Zobacz więcej informacji*. Lista klientów jest przewijana (stronicowana). Ponadto, powyżej, znajduje się wyszukiwarka klientów, gdzie po numerze Id, numerze telefonu, imieniu i nazwisku portier może szybko znaleźć klienta.
- b. Po kliknięciu *Zobacz więcej informacji* wyświetlają się szczegóły klienta - imię i nazwisko, dane kontaktowe takie jak email, numer telefonu.
 - i. Jeśli klient nie posiada żadnego aktywnego karnetu to wówczas wyświetli się informacja, aby aktywować karnet (trzeba go wybrać z listy karnetów, jakie wykupił klient). Po aktywacji karnetu pojawią się dodatkowe informacje, zawarte w podpunktach ii, oraz iii, w zależności od tego, jaki typ karnetu został aktywowany.
 - ii. Jeśli klient ma aktywny karnet typu “Dniowy” to wówczas oprócz typu karnetu wyświetli się również jego data ważności.
 - iii. Jeśli klient ma aktywny karnet typu “Wejściowy” to wówczas oprócz typu karnetu wyświetli się również liczba pozostałych możliwych wejść klienta
- c. Jeśli klient ma aktywny karnet, to oprócz danych wyświetli się panel rejestracji wejść/wyjść klienta.
 - i. W panelu tym, o ile klient planuje wejść na siłownię, możemy kliknąć przycisk *Rozpocznij trening* - wówczas jego wejście zostanie zarejestrowane. Można klientowi również przydzielić szafkę (odpowiednim przyciskiem) - nie jest to jednak konieczne. Po dokonanych operacjach rejestracji wejścia, po kliknięciu *zamknij*, strona się odświeża - zaktualizowane dane wyświetlają się.
 - ii. Jeśli klient odbył trening i chciałby go ukończyć możemy kliknąć *Zakończ trening* - wówczas zostanie zarejestrowane wyjście klienta a przydzielona szafka zostanie zwolniona. Strona również i w tym przypadku zostanie odświeżona i zaktualizowane dane wyświetlą się.

6. **Menedżer:**

- a. Po poprawnym zalogowaniu się, menedżer może łatwo i szybko wyświetlić wszystkie potrzebne mu do pracy statystyki sieci klubów fitness. Lista dostępnych statystyk jest widoczna w lewym panelu. Po wybraniu ich po prawej stronie wyświetli się dana statystyka.

12 Podręcznik administratora

Instrukcja uruchomienia aplikacji

Aby zainstalować naszą aplikację należy mieć na swoim komputerze zainstalowany Docker Desktop (<https://www.docker.com/products/docker-desktop/> [dostęp 19 stycznia 2024]). Po instalacji, należy go uruchomić, a następnie wykonać poniższe kroki.

1. Pobrać repozytorium naszej aplikacji: w terminalu należy wpisać komendę
``git clone https://gitlab.com/pzsp2_05/pzsp2_fitbit.git``
2. W terminalu przejść do lokalizacji pobranego repozytorium
3. Będąc w katalogu głównym projektu (/pzsp2_fitbit) należy utworzyć nowy plik o nazwie .env oraz treści:

```
DATABASE_PASSWORD = FitBit05x
```

```
EMAIL_HOST_PASSWORD = arpy jwnm iuuw txnb
```

Zawiera on hasła potrzebne do połączenia z bazą danych oraz wysyłania emaili

4. Nadal będąc w głównym katalogu projektu należy w terminalu wpisać komendę
``docker-compose build``
5. Po pomyślnym zakończeniu budowy projektu, aby uruchomić aplikację należy w terminalu wpisać komendę ``docker-compose up``
6. W przeglądarce należy przejść pod adres <http://localhost:3000/> - jest to nasza aplikacja
Aby przejść do aplikacji symulacji sprzętu należy w przeglądarce przejść pod adres <http://localhost:4000/>
7. Aby wyłączyć aplikację w terminalu, gdzie ją uruchomiliśmy należy wcisnąć na klawiaturze CTRL + C

Instrukcja uruchomienia testów jednostkowych

Wymagania są takie same jak przy uruchamianiu aplikacji, to znaczy należy mieć zainstalowanego Docker Desktop.

Należy również mieć pobrane repozytorium projektu i dodany plik .env (kroki 1, 2, 3 w instrukcji uruchomienia aplikacji)

Następnie wykonujemy poniższe kroki

1. Przechodzimy w terminalu do lokalizacji projektu (do katalogu głównego projektu - /pzsp2_fitbit)

2. W terminalu wpisujemy komendę ``docker build -t django_test -f backend/django_backend/Dockerfile.test .``
3. Po pomyślnym zakończeniu budowy w terminalu wpisujemy komendę ``docker run -it --rm django_test``
4. Widoczne będą wyniki testów jednostkowych oraz raport z narzędzia do pokrycia kodu

Zarządzanie bazą danych

Zarządzać bazą danych można przykładowo za pomocą SQL developer. Należy połączyć się z bazą używając

username: PZSP05

password: FitBit05x

hostname: ora2.ia.pw.edu.pl

port: 1521

service name: iaais

W repozytorium w katalogu sql/script znajdują się skrypty tworzące tabele w bazie , skrypty tworzące triggerzy oraz sekwencje, przykładowe inserty.

create_table.ddl - skrypt tworzący tabele

FitBit.sql - przykładowe inserty

sequence.sql - skrypty tworzące sekwencje

trigger.sql - skrypty tworzące triggerzy

13 Podsumowanie

Uważamy, że nasza aplikacja jest funkcjonalna i realizuje najważniejsze założenia. Jest podatna na rozwój dzięki rozbudowanej bazie danych i uniwersalności szaty graficznej. W kierunkach rozwoju przewidujemy dodawanie niezrealizowanych jeszcze funkcjonalności, które umożliwią użytkownikom korzystanie z aplikacji na jeszcze wyższym poziomie, m.in.: dodanie klientowi możliwości planowania własnych treningów, czy też implementacja algorytmów AI, które będą automatycznie układać plany treningowe zgodnie z preferencjami.

14 Bibliografia

Building Your first React Web Application, iMProgrammer, 2021 [dostęp 26 listopada 2023],

Dostępny w Internecie: <https://youtu.be/NzpbupWoIV4?si=unhK-ke7uOmeLJt3>

New to Figma? Get started with "Figma for beginners" tutorials, Figma, 2019 [dostęp 17 listopada 2023], Dostępny w Internecie:

https://www.youtube.com/watch?v=Cx2dkpBxst8&list=PLXDU_eVOJT7QHLShNqIXL1Cgbxj7HIN4&index=1

Podstawy UML, Marcin Pietraszek, 2020 [dostęp 6 listopada 2023], Dostępny w internecie:

<https://www.samouczekprogramisty.pl/podstawy-uml/>

Tutorial: Create and run your first GitLab CI/CD pipeline, [dostęp 22 listopada 2023], Dostępny w internecie: https://docs.gitlab.com/ee/ci/quick_start/

UX i UI design, co to jest i na czym polega praca UX designera?, Marta Kukuś, 2023 [dostęp 16 listopada 2023], Dostępny w Internecie:

<https://semcore.pl/ux-i-ui-design-co-to-jest-i-na-czym-polega-praca-ux-designera>

Dokumentacja biblioteki Reactstrap, dostępna w Internecie:

<https://reactstrap.github.io/?path=/story/home-installation--page>

Zatwierdzam dokumentację.	<div>.....</div> <div>.....</div> <div>Data i podpis Mentora</div>
---------------------------	--