**Temat projektu**

**Temat:** Aplikacja usprawniająca prace trenerów personalnych**.**

## Nazwa aplikacji: Sprevy

**Opis Projektu**

Aplikacja mobilna oraz webowa ma na celu usprawnienie pracy trenerów personalnych. W aplikacji będzie możliwość zdefinownia listy klientów wraz z dokumentacją dla danego klienta tak aby można było śledzić jego postępy. Dzięki aplikacji trener będzie miał możliwość ustalania swojego harmonogramu umawiając klinetów na poszczególne godziny (integracja z google calendar). Możliwość wysyłania planów treningowych oraz diet bezpośrednio za pomocą aplikacji dzięki czemu trener będzie miał również możliwość zdalnego prowadzenia podopiecznego. Aplikacja mobilna dla podopiecznych zawierająca panel klienta.

Technologie

Back-end

ASP.NET Core 3.1 framework odpowiedzialny za aplikację działającą na serwerze

MediatR – obsługa zapytań do serwera ściśle powiązana ze wzorcem CQRS

Baza danych - PostgreSQL – darmowa technologia relacyjnych baz danych

Wzorce projektowe

- Mediator - handlery odpowiedzialne za logikę biznesową do command i queries wywoływane w kontrolerach

- Command-query responsibility segregation (CQRS) - wzorzec odpowiedzialny za podział zapytań do serwera command - poleceń modyfikujących dane w dane oraz query - poleceń odpowiedzialnych za pobieranie danych

- Repository pattern - ułatwienie wprowadzania zmian w kodzie i jego utrzymania

- IoC- wstrzykiwanie zależności w klasach

Front-end

Angular 9 – framework webowy do tworzenia SPA(single page application) w Typescripcie

- RxJS – biblioteka do obsługi asynchronicznych operacji w apliakcji

- NGRX – implementacja wzorca projektowego Redux w angularze

- JWT – standard używany do obsługi sesji użytkownika z serwerem

Wzorce projektowe - opis wzorców które zostaną użyte w aplikacji frontendowej

-Observer – dostarczany przez bibliotekę RxJs do obsługi asynchronicznych części aplikacji (np. nasłuchiwanie na przyjście danych)

-Pamiątka – utrzymanie stanów obiektów obsługujących strumienie danych pochodzących z biblioteki RxJS (np. BehaviourSubject lub Subject)

-Singleton – natywnie w angularze do obsługi części aplikacji wspólnych dla różnych komponentów Wstrzykiwanie zależności – wzrorzec projektowy dostarczany przed Angulara pozwala na łatwe używanie potrzebnych zależności (np. serwisów) w klasach komponentów obsługujących widoki

- Interceptor – nasłuchiwanie na wystąpienie konkretnej operacji (np. wysłania zapytania do serwera) i przetwarzanie danych używanych w tej operacji jeszcze przed jej wykonaniem

Serwer

Aplikacja będzie hostowana na platformie Microsoft Azure przy użyciu Azure App Services. Rozwój aplikacji będzie wspierany przez platformę Azure Devops która przy uzyciu CI/CD będzie łączyła kod dodawany do repozytoriów z usługą App Service na której hostowana będzie aplikacja. App services będzie skonfigurowane z użyciem systemu Linux.

Funkcjonalności

- Wybór typu konta (widok startowy aplikacji) – widok z dwoma opcjami do wyboru „Trener” oraz „Klient” służący do wyboru w jakim trybie ma działać aplikacja: w trybie trenera czy w trybie klienta (tryby różnią się zestawem funkcjonalności) – po dokonaniu wyboru użytkownik jest przenoszony do panelu logowania

- Logowanie w trybie trenera – ten panel zawiera formularz logowania (który uwzględnia walidację wprowadzanych danych), gdzie należy podać adres email przypisany do konta oraz hasło, poniżej znajduje się przycisk „Zaloguj” (rozpoczynający operację logowania) oraz przycisk z napisem: „Nie masz jeszcze konta? Zarejestruj się!” przenoszący użytkownika do panelu rejestracji jako trener

- Możliwość utworzenia konta dla trenera – widok z formularzem rejestracyjnym (który uwzględnia walidację wprowadzanych danych) na którym należy podać: imię, nazwisko, adres email, hasło, oraz potwierdzić hasło), pod formularzem znajduje się przycisk „Zarejestruj”. Po udanej rejestracji użytkownik jest przenoszony do panelu z listą obsługiwanych klientów.

- Lista klientów dla trenera – widok zawierający listę obsługiwanych klientów przez trenera, po kliknięciu na klienta z listy pojawia się pop-up zawierający informacje na temat wybranego klienta: podstawowe dane (imię, nazwisko, email), staż treningu (w miesiącach), plan treningowy (kliknięcie tej pozycji przenosi użytkownika do widoku planu treningowego), prowadzoną dietę (jeśli posiada (kliknięcie tej pozycji przenosi użytkownika do widoku diety))

- Moduł odpowiadający za tworzenie planów treningowych, współdzielenie ich z klientami oraz wyświetlanie ich (tylko tryb trenera)– widok zawierający listę wcześniej stworzonych planów treningowych (kliknięcie pojedynczego planu przenosi użytkownika do widoku planu treningowego)

- Podgląd planu treningowego – widok zawierający opis planu treningowego (w formie pop-upa), oprócz opisu planu, znajduje się tutaj także pole, w którym po wpisaniu adresu email klienta można udostępnić plan treningowy danemu klientowi

- Moduł odpowiadający za tworzenie diet i współdzieleniu ich z klientami - widok zawierający opis tygodniowej diety (w formie pop-upa), oprócz opisu diety, znajduje się tutaj także pole, w którym po wpisaniu adresu email klienta można udostępnić plan diety danemu klientowi

- Podgląd planu diety – widok zawierający opis diety (w formie pop-upa), oprócz opisu planu, znajduje się tutaj także pole, w którym po wpisaniu adresu email klienta można udostępnić plan treningowy danemu klientowi

- Logowanie w trybie klienta – ten panel zawiera formularz logowania (który uwzględnia walidację wprowadzanych danych), gdzie należy podać adres email przypisany do konta oraz hasło, poniżej znajduje się przycisk „Zaloguj” (rozpoczynający operację logowania) oraz przycisk z napisem: „Nie masz jeszcze konta? Zarejestruj się!” przenoszący użytkownika do panelu rejestracji jako klient

- Możliwość utworzenia konta dla klienta – widok z formularzem rejestracyjnym (który uwzględnia walidację wprowadzanych danych) na którym należy podać: imię, nazwisko, adres email, hasło, oraz potwierdzić hasło), pod formularzem znajduje się przycisk „Zarejestruj”. Po udanej rejestracji użytkownik jest przenoszony do panelu z wykresem postępów klienta.

- Wykres postępów klienta – widok zawierający wykres na którym są przedstawione dane na temat częstotliwości treningów (na podstawie danych z kalendarza).

- Moduł odpowiadający za dodawanie treningów z klientami do kalendarza oraz za integrację z google calendar – widok w formie kalendarza na którym planowane będą treningi klienta. Na każdy z treningów w kalendarzu można kliknąć i na wyświetlonym pop-upie zobaczyć szczegóły treningu: datę, czy został zrealizowany (checkbox), odnośnik do planu treningowego klienta, odnośnik do diety jeśli klient realizuje dietę

- Monitorowanie przestrzegania diety przez klientów w celu podniesienia motywacji – widok dostępny dla klienta oraz trenera (do odczytu) gdzie klient w osobnym kalendarzu zaznacza czy danego dnia udało mu się zrealizować dietę (jest to dobrowolne, ale bazujemy na tym że osoby korzystające z aplikacji będą chciały pilnować same siebie pod tym kątem i dzielić się informacjami o diecie z klientem)

Lista zadań

* Stworzenie projektu w Azure Devops dodanie członków zespołu, nadanie uprawnień oraz utworzenie repozytorium Git – 1 d
* Stworzenie oraz konfiguracja projektu frontend – 0.5 d
* Stworzenie oraz konfiguracja projektu backend + baza danych – 1 d
* Opracowanie projektu architektury systemu – 1 d (cały zespół)
* Stworzenie mockupu widoku wyboru typu konta – 1 d
* Stworzenie mockupu rejestracji konta w trybie trener – 1 d
* Stowrzenie mockupu logowania do w trybie trener– 0.5 d
* Implementacja encji domenowych – 1 d
* Implementacja Handlera odpowiedzialnego za rejestracje użytkownika w trybie trener – 1 d
* Implementacja widoku rejestracji w trybie trener – 1d
* Implementacja Handlera odpowiedzialnego za logowanie użytkownika w trybie trener – 1 d
* Stworzenie widoku logowania w trybie trener – 1d
* Stworzenie mockupu dashbordu trenera – 0,5 d
* Implementacja CRUD dla encji ‘Klient’ – 1d
* Implementacja widoku dashbordu trenera - 1d
* Stworzenie mockupu widoku diety – 0.5 d
* Implementacja CRUD dla ‘Dieta’ – 1d
* Implementacja widoku diety – 1d
* Implementacja mechanizmu wysyłania diety -1d
* Implementacja pop-up diety - 4h
* Stworznie mockupu widoku planu treningowego - 0.5 d
* Implementacja CRUD dla Planu trenigowego – 1d
* Implementacja widoku planu treningowego – 1d
* Implementacja mechanizmu wysyłania planu do konkretnego użytkownika – 3h
* Implementacja pop-up planu treningowego – 5h
* Stworzenia mockup dla panelu logowania w trybie klient – 5h
* Implementacja Handlera odpowiadającego za logowanie klienta – 1d
* Implementacja widoku panelu logowania klienta – 1d
* Implementacja mockup dla panelu rejestracji w trybie klient -1d
* Implementacja Handlera odpowiedzialnego za rejestracje klienta – 1d
* Implementacja widoku panelu rejestracji w trybie klient - 1d
* Stworzenie mockup dla panelu statystyk klienta – 1d
* Implementacja klasy odpowiedzialnej za kalkulacje statystyk – 1d
* Implementacja Handlera odpowiedzialnego za agregacje statystyk – 1d
* Implementacja widoku panelu statystyk -1d
* Integracja z google calendar research – 1d
* Integracja z google calendar - 1d
* Stworzenie mockup panelu monitorowania przestrzegania diety - 1d
* Implemntacja klasy odpowiedzialnej za kalkulacje statystyk – 1d
* Implementacja Handlera odpowiedzialnego za pobranie statystyk dotyczących przestrzegania diety -1d
* Implementacja widoku panelu przestrzegania diety – 1d