

# Aplikacja do Monitorowania Postępów Fitness -

Maciej Szewczuk, Adam Stankiewicz

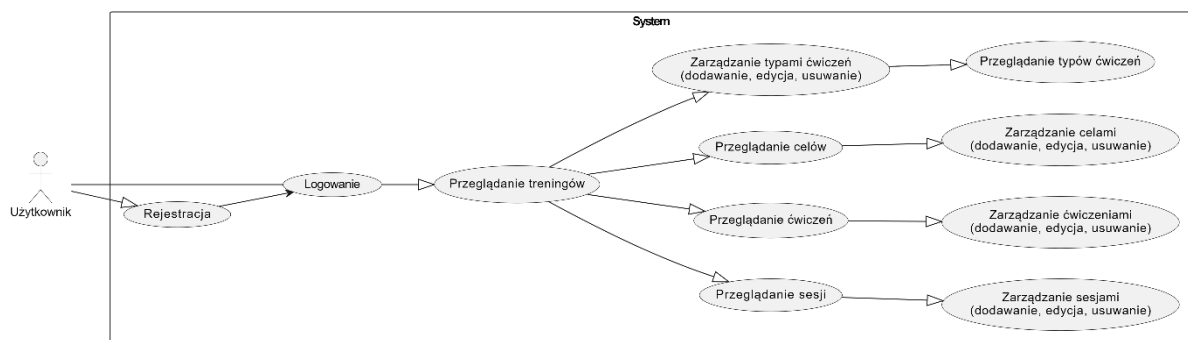
## 1. Wprowadzenie

Aplikacja do Monitorowania Postępów Fitness została stworzona w technologii ASP.NET, wykorzystując Entity Framework (EF) do obsługi ORM. Celem aplikacji jest umożliwienie użytkownikowi zapisywania osiągniętych wyników treningów na siłowni. Użytkownik może dodawać sesje treningowe, składające się z wybranych ćwiczeń oraz zapisywać wyniki, takie jak obciążenie, liczba serii i liczba powtórzeń. Dodatkowo, aplikacja dostarcza użytkownikowi statystyki, prezentujące liczbę sesji treningowych w ostatnich czterech tygodniach oraz najlepsze wyniki dla każdego rodzaju ćwiczenia.

## 2. Struktura Projektu

Projekt składa się z warstw interfejsu użytkownika, warstwy biznesowej oraz warstwy dostępu do danych. ASP.NET odpowiada za obsługę żądań HTTP, a EF jest wykorzystywane do interakcji z bazą danych.

### 2.1 Diagram przypadku użycia



## 3. Baza Danych

### 3.1 Modele

#### 3.1.1 Session

- **Id** - identyfikator sesji treningowej.
- **Start** - data rozpoczęcia sesji.
- **End** - data zakończenia sesji.
- **UserId** - identyfikator użytkownika powiązanego z sesją.
- **SessionName** - nazwa sesji treningowej

### 3.1.2 ExerciseType

- **Id** - identyfikator rodzaju ćwiczenia.
- **Name** - nazwa rodzaju ćwiczenia.
- **UserId** - identyfikator użytkownika powiązanego z rodzajem ćwiczenia.

### 3.1.3 Exercise

- **Id** - identyfikator ćwiczenia.
- **Weight** - obciążenie w kilogramach.
- **Reps** - liczba powtórzeń w serii.
- **Series** - liczba serii.
- **ExerciseTypeId** - identyfikator rodzaju ćwiczenia.
- **SessionId** - identyfikator sesji treningowej.
- **UserId** - identyfikator użytkownika powiązanego z ćwiczeniem.

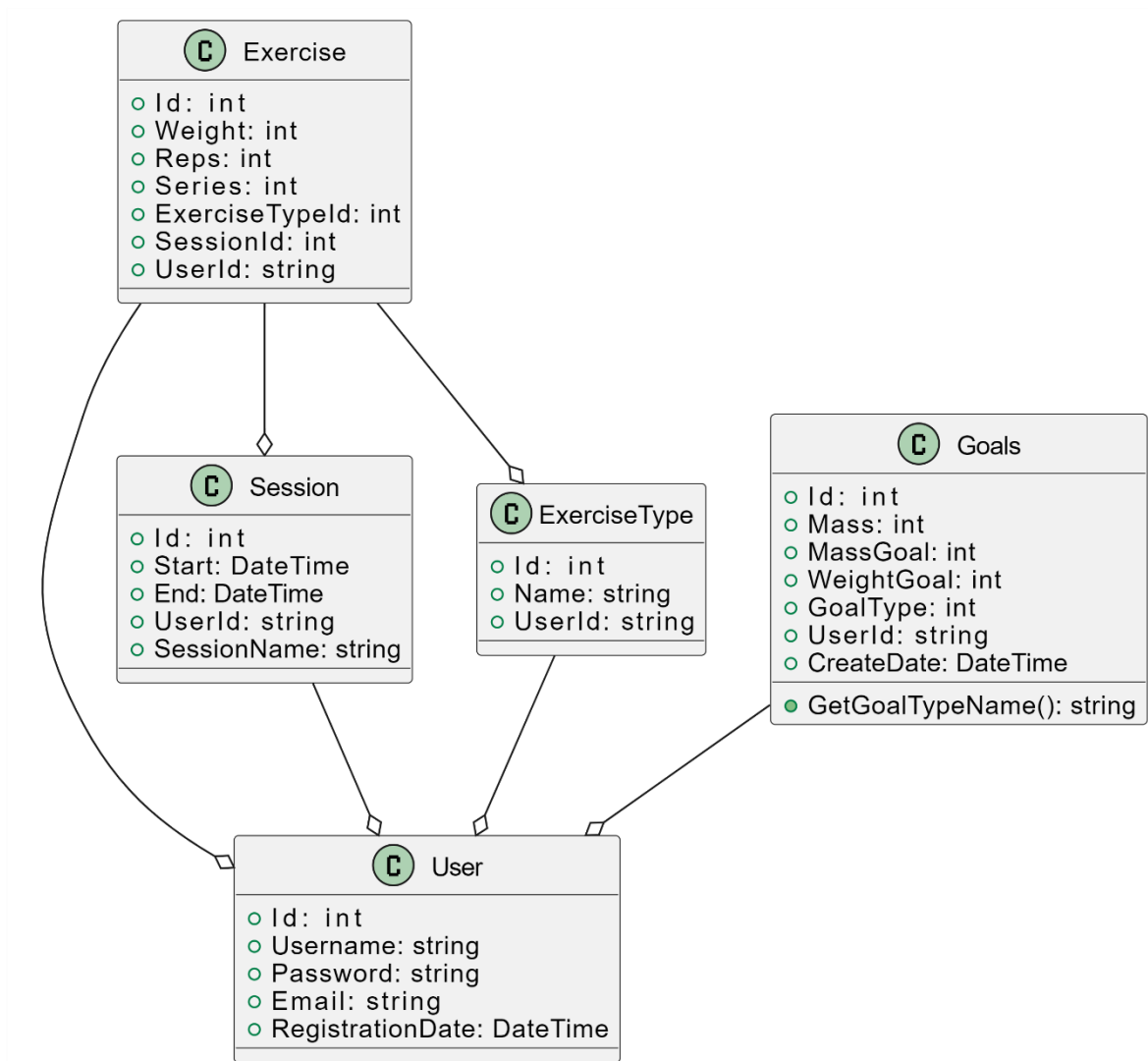
### 3.1.4 Goals

- **Id**: identyfikator celu.
- **Mass**: aktualna masa ciała użytkownika.
- **MassGoal**: docelowa masa ciała użytkownika.
- **WeightGoal**: docelowa waga użytkownika.
- **GoalType**: typ celu (1 - Podnieść Ciężar, 2 - Utrata wagi).
- **UserId**: identyfikator użytkownika powiązanego z celem.
- **User**: obiekt klasy **IdentityUser** reprezentujący użytkownika.
- **CreateDate**: data utworzenia celu (domyślnie ustawiona na aktualny czas).
- **GetGoalTypeName()**: metoda zwracająca nazwę typu celu na podstawie wartości pola **GoalType**.

### 3.2 Relacje

- Relacja między **Session** a **User** - Many-to-One.
- Relacja między **ExerciseType** a **User** - Many-to-One.
- Relacja między **Exercise** a **ExerciseType** - Many-to-One.
- Relacja między **Exercise** a **Session** - Many-to-One.

### 3.3 Diagram klas



## 4. Funkcjonalność

### 4.1 Rejestracja i Logowanie

Aplikacja umożliwia użytkownikom rejestrację nowych kont oraz logowanie. Każdy użytkownik ma unikalny identyfikator.

### 4.2 Widoki Publiczne

Dostępne dla wszystkich użytkowników, obejmują listę przykładowych treningów.

### 4.3 Widoki Prywatne

Dostępne tylko dla zalogowanych użytkowników, umożliwiają monitorowanie postępów fitness, ustawianie celów i zapisywanie treningów. **Wszystkie wprowadzone sesje i ćwiczenia dostępne są tylko dla danego zalogowanego użytkownika, inni użytkownicy nie mają do nich dostępu.**

#### 4.4 Formularz Dodawania Nowych Treningów

Pozwala użytkownikom dodawać nowe sesje treningowe, gdzie dane są przypisane do zalogowanego użytkownika.

##### 4.4.1 Dodawanie Ćwiczeń

Użytkownik ma możliwość dodawania nowych ćwiczeń poprzez formularz, w którym podaje:

- Obciążenie (**Weight**) w kilogramach.
- Liczbę powtórzeń (**Reps**) w serii.
- Liczbę serii (**Series**).
- Rodzaj ćwiczenia (**ExerciseTypeId**), wybierając go spośród dostępnych typów ćwiczeń.
- Sesję treningową (**SessionId**), do której przypisane jest ćwiczenie.

##### 4.4.2 Edycja Ćwiczeń

Zalogowani użytkownicy mają możliwość edycji istniejących ćwiczeń. Proces edycji obejmuje zmianę wartości obciążenia, liczby powtórzeń, liczby serii, rodzaju ćwiczenia oraz sesji treningowej, do której ćwiczenie jest przypisane.

##### 4.4.3 Usuwanie Ćwiczeń

Użytkownicy mogą usuwać ćwiczenia, co powoduje trwałe usunięcie zapisanych danych dotyczących danego ćwiczenia z bazy danych.

##### 4.4.4 Dodawanie Typów Ćwiczeń

Użytkownik ma możliwość dodawania nowych typów ćwiczeń poprzez formularz, w którym podaje nazwę (**Name**) nowego rodzaju ćwiczenia.

##### 4.4.5 Edycja Typów Ćwiczeń

Zalogowani użytkownicy mają możliwość edycji istniejących typów ćwiczeń, zmieniając ich nazwę.

##### 4.4.6 Usuwanie Typów Ćwiczeń

Użytkownicy mogą usuwać typy ćwiczeń, ale nie można tego zrobić, jeśli istnieją ćwiczenia przypisane do danego typu. W takim przypadku konieczne jest najpierw usunięcie lub przeniesienie ćwiczeń do innego rodzaju ćwiczenia.

##### 4.4.7 Dodawanie Sesji Treningowych

Zalogowani użytkownicy mają możliwość dodawania nowych sesji treningowych poprzez formularz, w którym podają:

- Datę rozpoczęcia (**Start**) sesji.
- Datę zakończenia (**End**) sesji.
- Nazwę sesji (**SessionName**).
- Wybrane ćwiczenia i ich wyniki.

#### 4.4.8 Edycja Sesji Treningowych

Użytkownicy mogą edytować dane sesji treningowej, takie jak data rozpoczęcia, data zakończenia, nazwa sesji oraz przypisane ćwiczenia wraz z wynikami.

#### 4.4.9 Usuwanie Sesji Treningowych

Usuwanie sesji treningowych skutkuje również usunięciem przypisanych do niej ćwiczeń i ich wyników. Użytkownicy mają możliwość trwałego usunięcia sesji treningowej z bazy danych.

#### 4.5 Zarządzanie Celami

Użytkownicy mogą dodawać, edytować i usuwać cele związane z treningiem. Każdy cel ma określoną masę ciała, docelową masę, docelową wagę oraz typ celu (Podnieść Ciężar lub Utrata wagi). Dodatkowo, użytkownicy mają dostęp do statystyk związanych z postępem w realizacji celów.

### 5. Dane techniczne

#### 5.1 Technologie

- **Framework ASP.NET:** Wykorzystano ASP.NET do budowy warstwy prezentacji i obsługi żądań HTTP.
- **Entity Framework (EF):** Użyto EF do obsługi ORM (Object-Relational Mapping) i dostępu do bazy danych.
- **C#:** Język programowania używany do implementacji logiki biznesowej i obsługi zdarzeń na serwerze.
- **HTML, CSS, JavaScript:** Technologie używane w warstwie interfejsu użytkownika.

#### 5.2 Baza Danych

- **System Zarządzania Bazą Danych (DBMS):** Zastosowano Microsoft SQL Server lub inny odpowiedni system do przechowywania danych.
- **Baza Danych**

#### 5.3 Framework do Interfejsu Użytkownika

- **Framework CSS/JavaScript (np. Bootstrap, jQuery):** Używane do responsywnego projektowania i obsługi interakcji po stronie klienta.

#### 5.4 Bezpieczeństwo

- **Autentykacja i Autorozpoznawanie:** Zastosowano system logowania oparty na indentyfikacji użytkownika.

#### 5.5 Zarządzanie Projektem

- **System Kontroli Wersji:** Wskazanie, czy projekt był zarządzany przy użyciu systemu kontroli wersji (np. Git).

## 6. Uruchomienie projektu

Instrukcja uruchomienia projektu:

1. Zainstaluj git for Windows oraz Visual Studio
2. Klikając prawym przyciskiem w Explorze Windows wybierz opcję otwórz w terminalu i wpisz następującą komendę

git clone [https://github.com/insreed/projekt\\_asp.net.git](https://github.com/insreed/projekt_asp.net.git)

4. Jeśli w folderze Data/Migrations jest jakikolwiek plik - usuń go.
5. W Visual studio po wczytaniu projektu kliknij na narzędzia -> Menadżer pakietów NuGet -> Konsola menadżera pakietów
6. Wklej kolejno polecenia:

Add-Migration 001 -OutputDir "Data/Migrations"

Update-Database

8. Uruchom projekt.