

Count Calories

Piotr Przybyła 259286, Radosz Werner 259310

06.05.2023

Spis treści

1	Wstęp	2
2	Opis kodu źródłowego	2
2.1	WPF	2
2.2	Entity Framework	2
2.3	API	3
3	Opis działania aplikacji	3
3.1	Okno główne - Count Calories	3
3.2	Okno AddMeal	3
3.3	Okno AddIngredient	3
3.4	Okno Add Ingredient from list	3
4	Podsumowanie	4

1 Wstęp

Aplikacja Count Calories jak sama nazwa wskazuje służy do liczenia kalorii. Użytkownik może dodać posiłek. Każdy posiłek składa się ze składników, które można dodać ręcznie lub wyszukać w API: <https://api-ninjas.com/api/nutrition>

2 Opis kodu źródłowego

2.1 WPF

Używamy WPF do wyświetlania okienek aplikacji - mamy 4 okna:

- MainWindow - Główne okno aplikacji, służy do wyświetlania posiłków, oraz ma przyciski do edycji, usuwania i dodawania posiłków;
- AddMealWindow - Okno służące zarówno do dodawania nowego posiłku jak i edycji istniejącego - zawiera listę składników oraz 2 przyciski - jeden do dodania składnika i drugi do zatwierdzenia posiłku;
- AddIngredient - Okno służące do dodania składnika do posiłku - zawiera inputy do wpisywania wartości kalorii i makroskładników. Posiada również przycisk do zatwierdzenia oraz przycisk do dodania produktu z API;
- AddFromApi - Okno do dodawania produktu z API - ma pole tekstowe do wpisania nazwy oraz przycisk który wywołuje zapytanie do API;

Wszystkie okna zawierają metody obsługujące wyświetlenie elementów UI, reagujące na zmiany w edytowalnych polach tekstowych oraz obsługujące kliknięcie przycisków.

2.2 Entity Framework

Baza danych zawiera 3 encje:

- Meal - encja przechowująca wszystkie posiłki;
- Product - encja przechowująca dane na temat produktu - kalorie i makroskładniki;
- Ingredient - encja tworząca relację między Meal i Product oraz zawierająca informację o wadze danego produktu w posiłku;

Wszystkie encje posiadają metody obsługujące CRUD.

2.3 API

Korzystamy z API które przyjmuje nazwę produktu oraz zwraca kalorie oraz makroskładniki tego produktu. Po kliknięciu odpowiedniego przycisku i wpisaniu nazwy produktu wysyłane jest zapytanie GET do API.

3 Opis działania aplikacji

3.1 Okno główne - Count Calories

Gdy uruchamiamy aplikację przed naszymi oczami pojawia się pusta lista posiłków. Jesteśmy w stanie dodać nowy posiłek, edytować obecny (jeśli jest na liście), usunąć dodany posiłek. W celu edycji lub usunięcia posiłku musimy wybrać go z listy, a następnie użyć odpowiedniego przycisku.

3.2 Okno AddMeal

Gdy zdecydujemy się dodać lub edytować posiłek, pojawi się okno AddMeal. Widnieje w nim textbox z nazwą posiłku, które musimy uzupełnić bezpośrednio przed jego akceptacją. Mamy możliwość dodawania składników przyciskiem AddIngredient, co otworzy kolejne okno. Po dodaniu odpowiedniej ilości składników dodajemy posiłek do listy przyciskiem.

3.3 Okno AddIngredient

Tutaj mamy 2 możliwości. Albo odpowiednio uzupełniamy pola tekstowe makrosami naszego produktu i dajemy wagę składnika, albo za pomocą przycisku "Add ingredient from list" Otwieramy kolejne okno, umożliwiając dodanie produktu z API, pola tekstowe uzupełnią się, zostanie tylko waga. Makro produktu zostanie odpowiednio przeliczone zależnie od wagi. Po uzupełnieniu wszystkich pól akceptujemy składnik przyciskiem.

3.4 Okno Add Ingredient from list

Proste okno umożliwiające dodanie produktu z API. Wpisujemy nazwę interesującego nas produktu, dla testu np. "bread" lub "cheese" i akceptujemy przyciskiem. Wyświetli się okno AddIngredient uzupełnione o makro danego produktu.

4 Podsumowanie

Projekt wiele nas nauczył. Opanowaliśmy tworzenie baz danych z wykorzystaniem dbLocal i EntityFramework 6, nauczyliśmy się programować w języku C# (był to nasz pierwszy projekt w tej technologii). Pisanie kodu sprawiło nam trochę problemów, wynikały one przede wszystkim z złego rozplanowania bazy danych, jednak po poprawkach wszystko się trzyma, gra i buczy. Projekt z naszej strony uważamy za udany.