

# **Platformy programistyczne .Net i Java**

## **Planer z API pogody**

Wtorek 17:05-18:45 Y02-41b

Rafał Marecki 252901

Michał Słowiński 254018

22 kwietnia 2022

## Spis treści

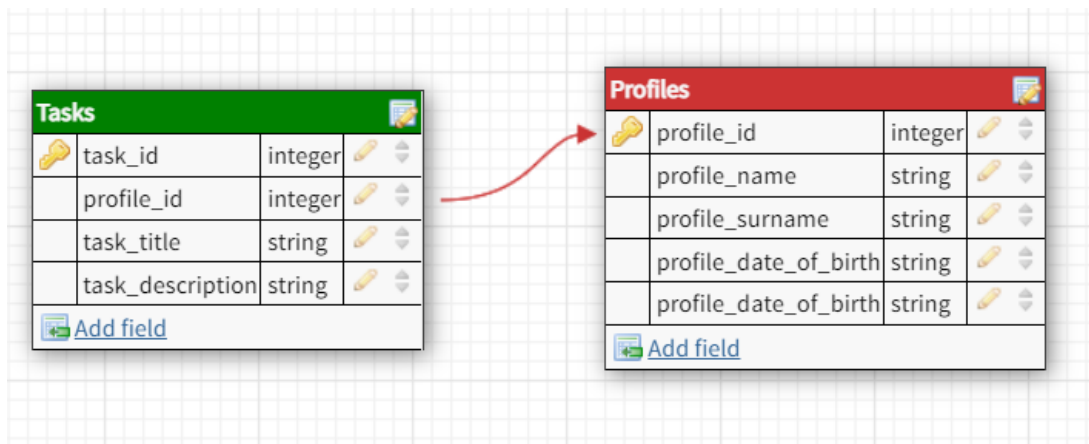
1	Opis projektu	3
2	Baza danych	3
3	Użyte API	4
4	Prezentacja graficzna aplikacji	5
5	Podsumowanie i wnioski	7

## 1 Opis projektu

Aplikacja planer została napisana w języku C w Entity Framework. Jest to projekt WPF (Windows Presentation Foundation), z podłączoną bazą danych ORM, co umożliwia presystencję danych w aplikacji klienckiej po jej zamknięciu. Użytkownik może swobodnie wpisywać dane do aplikacji, tworzyć nowe zadania, lub edytować i usuwać już istniejące.

## 2 Baza danych

W bazie danych znajdują się dwie tabele: Profiles i Tasks. Tabela Profiles łączy się z tabelą Tasks relacją jeden do wielu, a kluczem obcym w tabeli Tasks jest kolumna profile\_id.



Rysunek 1: Poglądowy schemat bazy danych

```
CREATE TABLE [dbo].[Tasks] (
    [task_id] INT IDENTITY (1, 1) NOT NULL,
    [profile_id] INT NOT NULL,
    [task_title] NVARCHAR (50) NOT NULL,
    [task_description] NVARCHAR (MAX) NULL,
    [task_date] DATE NOT NULL,
    [task_time_start] TIME (7) NULL,
    [task_time_end] TIME (7) NULL,
    PRIMARY KEY CLUSTERED ([task_id] ASC),
    CONSTRAINT [FK_Tasks_ToProfile] FOREIGN KEY ([profile_id]) REFERENCES [dbo].[Profiles] ([profile_id]) ON DELETE CASCADE
);

CREATE TABLE [dbo].[Profiles] (
    [profile_id] INT IDENTITY (1, 1) NOT NULL,
    [profile_name] NVARCHAR (50) NOT NULL,
    [profile_surname] NVARCHAR (50) NOT NULL,
    [profile_date_of_birth] DATE NOT NULL,
    PRIMARY KEY CLUSTERED ([profile_id] ASC)
);
```

Rysunek 2: Kod tabel

### 3 Użyte API

Zdecydowaliśmy się na użycie API OpenWeather. Jest to darmowe API do sprawdzania pogody. Po zalogowaniu się na stronie producenta i wygenerowaniu klucza, mogliśmy użyć go w celu otrzymania informacji o warunkach pogodowych w danym mieście.

Informacje te były otrzymywane w formacie JSON, zostały zdeserializowane i wprowadzone do klasy WeatherInformation. Informacje ważne dla działania naszej aplikacji to informacje o temperaturze, opisie słownym pogody wraz z odpowiadającą mu ikoną i ciśnieniu danego dnia. Domyślnym miastem, dla którego aplikacja wyświetla pogodę jest Wrocław, ale użytkownik ma możliwość zmiany miasta. Należy pamiętać, że API jest anglojęzyczne, dlatego należy wpisywać angielskie wersje miasto np. Wrocław to Wrocław, ponieważ to API nie przyjmuje polskich znaków



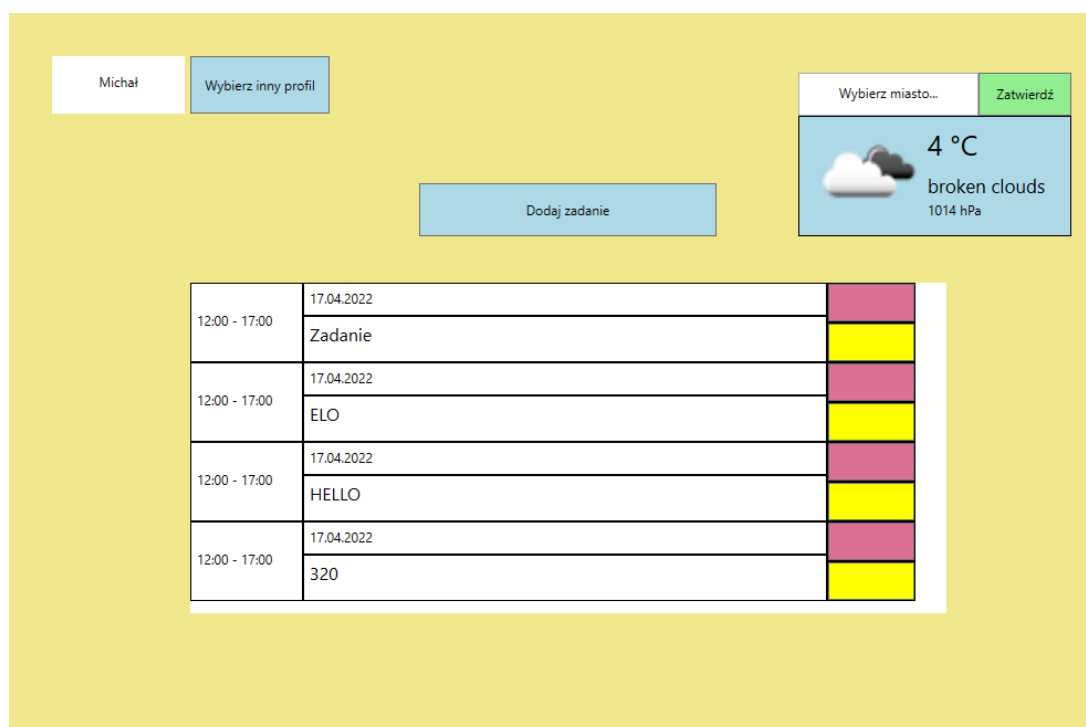
Rysunek 3: Prezentacja pogody w aplikacji

## 4 Prezentacja graficzna aplikacji

Aplikacja posiada główne okno, na którym widać informację o pogodzie w wybranym mieście (domyślnie Wrocław), przewijalną listę zadań (domyślnie brak zadań), przycisk służący do dodania zadania oraz przycisk do zmiany profilu.

Zarówno przycisk dodania zadania, jak i przycisk zmiany profilu otwiera nowe okno, w którym użytkownik może dodać zadanie, wraz z jego wszystkimi atrybutami, lub zmienić (albo dodać nowego) użytkownika.

Jeśli jakieś zadania są dodane do listy, użytkownik ma opcję usunięcia, lub edycji danego zadania za pomocą przycisku. Czerwony przycisk służy do usuwania, a żółty do edycji.



Rysunek 4: Strona główna aplikacji

UTWÓRZ NOWE ZADANIE

Nazwa zadania:

Data zadania:

Początek:  Koniec:

Opis zadania:

Zatwierdź

Rysunek 5: Strona dodawania nowego zadania

WYBIERZ PROFIL

Michał  
Rafał  
Tomasz  
Jakub  
Radosław  
Radosław  
Dawid  
Adam  
Robert  
Michał  
Jacek  
Kuba

lub

UTWÓRZ NOWY PROFIL

Imię:

Nazwisko:

Data urodzenia:

Zatwierdź

Zatwierdź

Rysunek 6: Strona wyboru/ dodania profilu

## 5 Podsumowanie i wnioski

- Udało się stworzyć aplikację planera w C z użyciem Entity Framework
- Stworzono interfejs użytkownika wykorzystujący technologię WPF
- Dane wpisane przez użytkownika pozostają w aplikacji nawet po jej zamknięciu
- Użyto bazy danych ORM
- Zastosowano połączenie sieciowe z API OpenWeather do sprawdzania prognozy pogody.
- Dane otrzymane od API w formacie JSON zdeserializowano i wpisano do klasy
- Dane zostały zaprezentowane graficznie w postaci tabelki z użyciem ScrollViewerea i StackPaneli

Udało się spełnić podstawowe założenia aplikacji, niestety brakuje dokumentacji i testów jednostkowych.