



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

Dokumentacja projektu

Przedmiot: Programowanie

Tytuł projektu: MVVM Weather App

Prowadzący:

Dr Marek Jaszuk

Wykonawca: *Oskar Kogut*

Grupa: *2-IID-ZP/SW02*

Rzeszów, 2019

1. Opis założeń projektu

Aplikacja może zostać wystawiona na sprzedaż w sklepie windows. Potencjalni użytkownicy szukający aplikacji wyświetlającej pogodę mogą ją zakupić.

2. Specyfikacja wymagań

Wymagania funkcjonalne:

- Aplikacja powinna pokazywać aktualną prognozę pogody na 5 kolejnych dni w wybranym przez użytkownika mieście
- W przypadku braku połączenia z internetem aplikacja powinna wyświetlać odpowiedni komunikat

Wymaganie niefunkcjonalne:

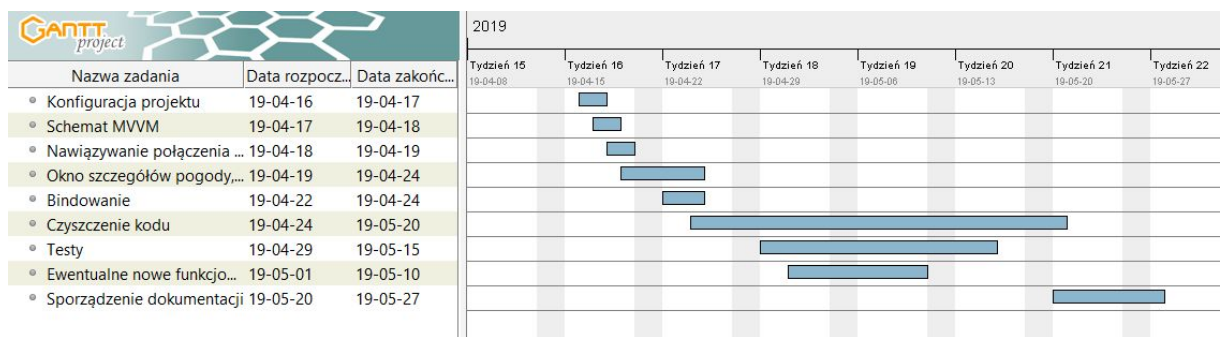
- Aplikacja powinna działać płynnie
- Aplikacja nie powinna wyłączać się bez powodu
- Aplikacja nie powinna zużywać więcej, niż 128MB pamięci RAM

3. Diagram przypadków użycia



Oskar Kogut, dokumentacja projektu "MVVM Weather App"

4. Harmonogram realizacji projektu



5. Opis techniczny projektu

Aplikacja została zrealizowana w oparciu o wzorzec projektowy MVVM (model, view, view-model).

Komponentami wzorca MVVM są:

- **Model** - warstwa odpowiadająca za dane aplikacji

Modelem w omawianej aplikacji są pliki

[AccuWeatherCity.cs](#), oraz [AccuWeatherForecast.cs](#).

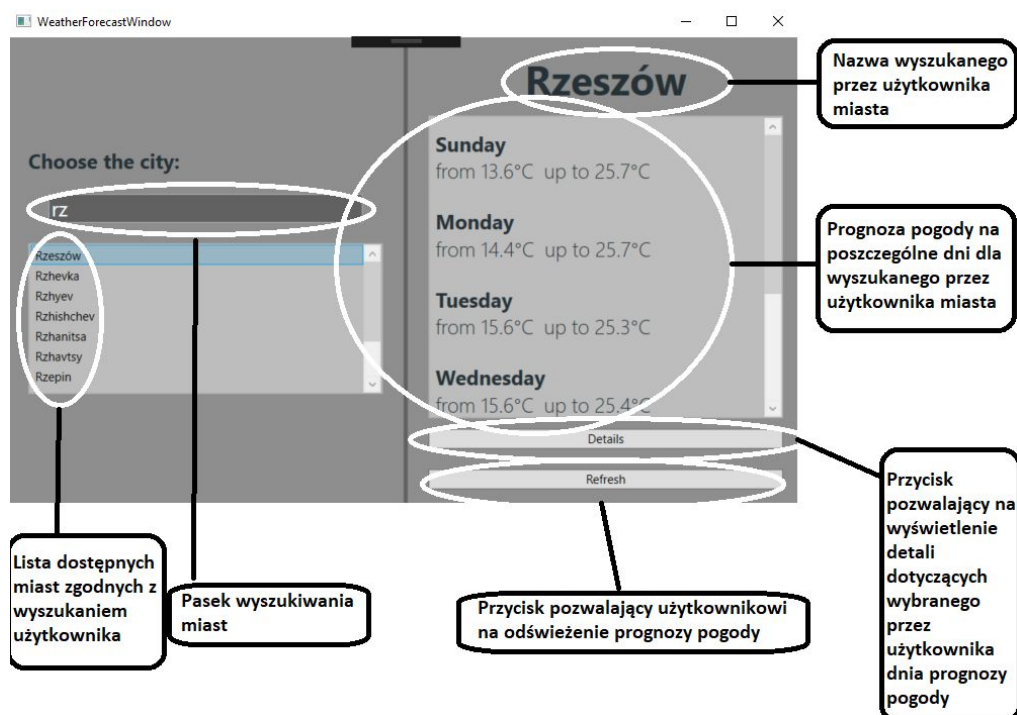
Oba pliki zawierają właściwości (properties) pozwalające na deserializację do nich obiektów JSON uzyskanych za pomocą zapytania Rest API.

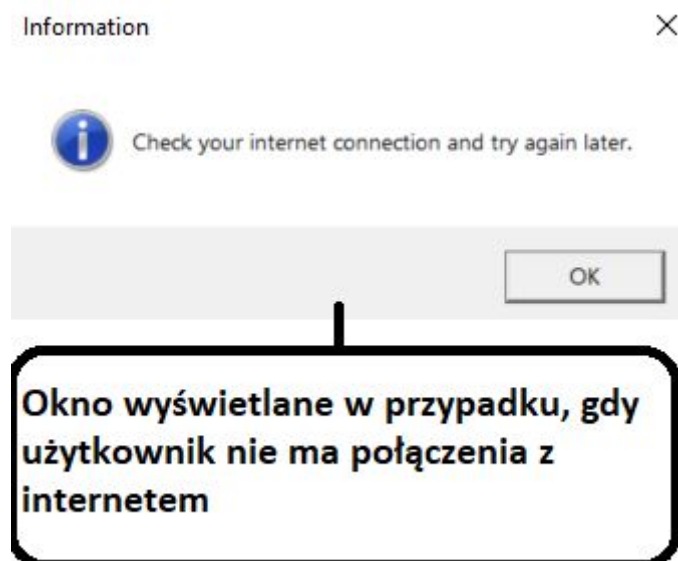
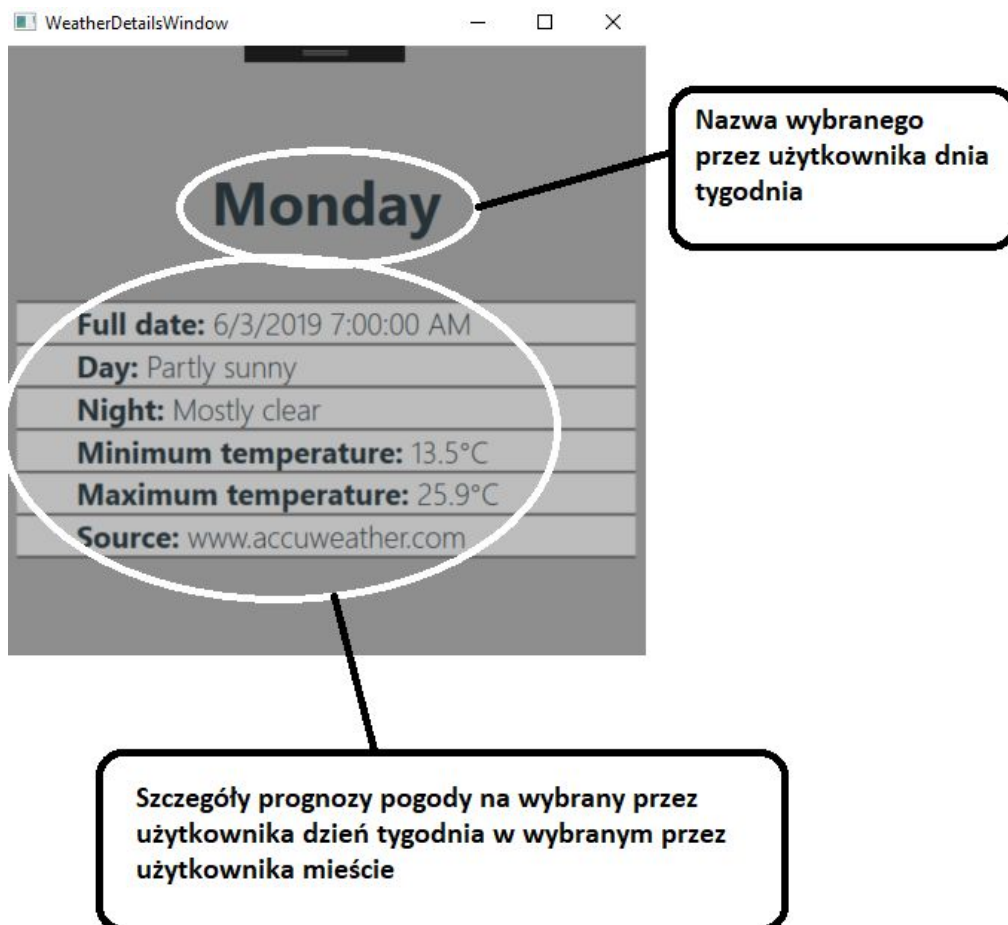
- **View** - warstwa odpowiadająca za widoczne dla użytkownika komponenty aplikacji

W przypadku warstwy widokowej, w omawianej aplikacji są to okna: [WeatherDetailsWindow](#), oraz [WeatherForecastWindow](#). Pozwalają one na interakcję użytkownika z programem poprzez renderowanie widocznych dla niego okienek.

- View-Model** - Odpowiada za warstwę logiczną aplikacji. W omawianej aplikacji, za warstwę view-model odpowiadają klasy [WeatherAPI.cs](#), oraz [WeatherVM.cs](#). [WeatherAPI.cs](#) zawiera metody pozwalające na komunikację aplikacji z AccuWeather API. [WeatherVM.cs](#) pozwala na bindowanie przez View odpowiednich obiektów stworzonych na podstawie metod z [WeatherAPI](#). W view-model są też [DetailsCommand](#) i [RefreshCommand](#) - implementujące interfejs [ICommand](#) klasy posiadające właściwości odpowiedzialne za przyciski.
- Binding** - synchronizuje **View Model** z **View**. W kodzie XAML okien [WeatherDetailsWindow](#), oraz [WeatherForecastWindow](#) używamy bindowania do aktualizowania odpowiednich okienek programu.

6. Prezentacja warstwy użytkowej projektu

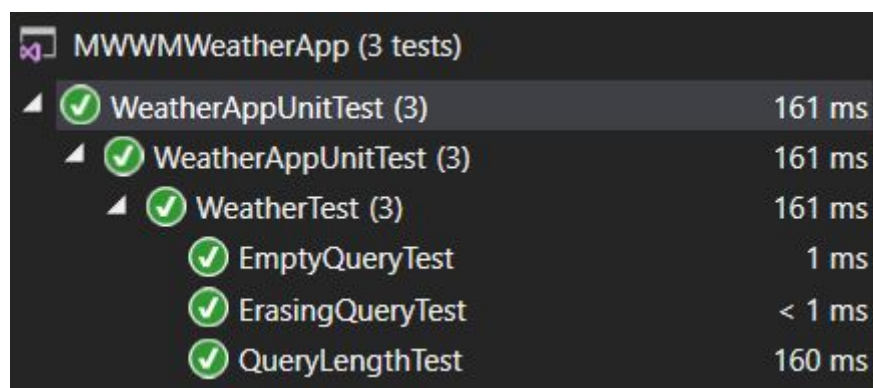




7. Link do repozytorium

<https://github.com/Kretoskar/wpf-mvvm-weather-app>

8. Zrzut ekranu z pozytywnie przeprowadzonymi testami.



9. Materiały źródłowe

Tutoriale:

1. **Eduardo Rosas** - Windows Presentation Foundation Masterclass
2. **Denis Panjuta** - Complete C# Masterclass
3. **Tod Vachev** - Basics of object oriented programming with C#

Literatura:

1. **Jennifer Greece, Andrew Stellman** - Head First C#
2. **Joseph Albahari, Ben Albahari** - C# 7.0 in a nutshell
3. **Robert C. Martin** - Czysty kod

Oskar Kogut, dokumentacja projektu "MVVM Weather App"

10. Dokumentacja

<https://drive.google.com/file/d/1SL4LLhcIQo-VwmJdJ4BR0ubUn0DnWN-B/view?usp=sharing>