

„weaRther”

Dokumentacja projektu z przedmiotu Mobilne Interfejsy Multimedialne

Gliwice, 04.04.2022r.

Autorka: Natalia Szalas

Spis treści

1. Opis projektu
2. Struktura plików
3. Interfejs graficzny
4. OpenWeatherMap API
5. Podsumowanie

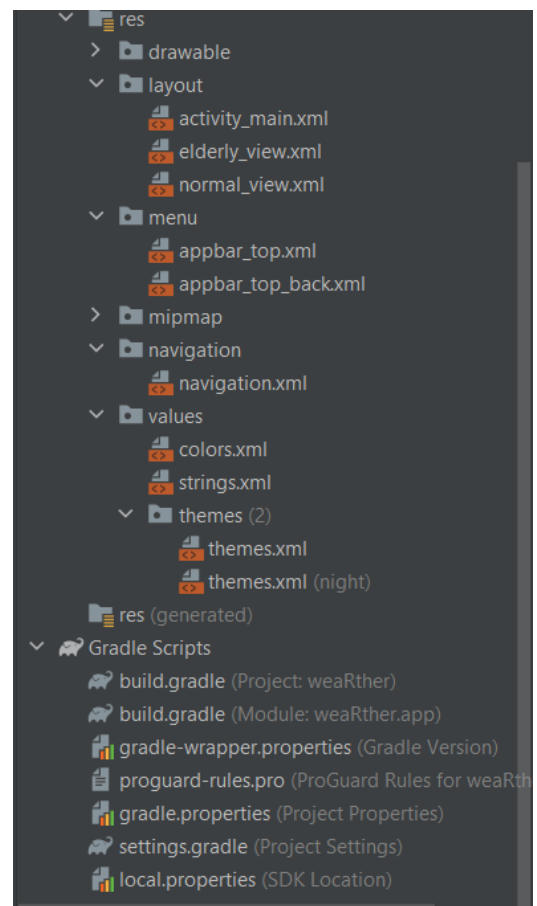
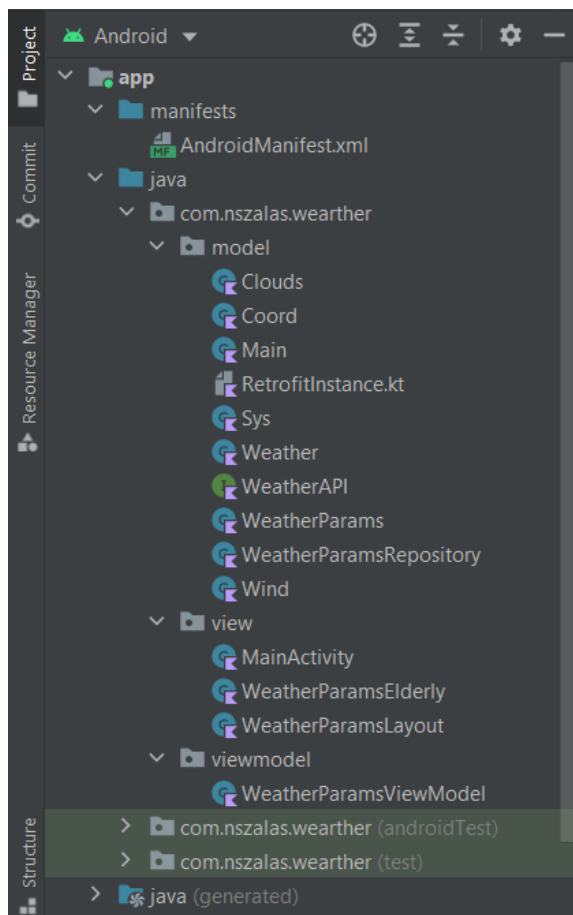
1. OPIS PROJEKTU

Głównym celem projektu było stworzenie aplikacji mobilnej, która pobiera z zewnętrznego API informacje o aktualnym stanie pogody. Funkcjonalność jest prosta, aczkolwiek przejrzysta oraz przemyślana pod kątem zwykłego użytkownika, jak i użytkowników starszych typu senior. Nie ogranicza się ona do konkretnej lokalizacji, dlatego posiada również możliwość wyszukania miejsca, o którym chcielibyśmy się więcej dowiedzieć. Aplikacja pobiera dane z API udostępnianego przez stronę openweathermap.org dla wszystkich miast na świecie.

Projekt powstał w języku Kotlin z zastosowaniem modelu MVVM (Model – View – ViewModel). Technologią wykorzystaną do jego stworzenia jest Android Studio wraz z różnymi bibliotekami i wtyczkami. Do tego interfejs graficzny został zaprojektowany zgodnie z zasadami Material Design, a ikony oraz tło powstały w aplikacji Canva.

Nazwa aplikacji powstała z angielskiego słowa 'weather' oznaczającego pogodę, jednakże wzbogacona w literę 'r' zmienia znaczenie. Wymowa fonetyczna brzmi wtedy 'wear there', co przetłumaczone zostać może na „ubierz tam”. Ma to odniesienie do przydatności aplikacji przy określeniu odpowiedniego stroju do aktualnych warunków atmosferycznych.

2. STRUKTURA PLIKÓW



3. INTERFEJS GRAFICZNY

Widok podstawowy aplikacji:



Widok dla seniorów:



4. OpenWeatherMap API

Do poprawnego działania aplikacji niezbędne jest zarejestrowanie się na stronie openweathermap.org w celu uzyskania klucza dostępu do API. Następnie wpisujemy go w pliku WeatherAPI.kt zaraz za znakami 'APPID='

```
1 package com.nszalas.weather.model
2
3 import ...
4
5
6
7
8 // "https://api.openweathermap.org/data/2.5/weather?q=Katowice&APPID=98ed72d7190f406b2e83af52b3dc15f7"
9
10 interface WeatherAPI {
11
12     @GET( value: "data/2.5/weather?&units=metric&APPID=98ed72d7190f406b2e83af52b3dc15f7")
13     fun getAllParams(
14         @Query( value: "q") name:String
15     ): Call<WeatherParams>
16
17 }
18 }
```

Program pobiera informacje z zewnątrz w postaci pliku JSON, np.:

```
{ "coord": { "lon": 19.0275, "lat": 50.2584 }, "weather": [ { "id": 804, "main": "Clouds",
"description": "overcast
clouds", "icon": "04n" } ], "base": "stations", "main": { "temp": 275.89, "feels_like":
274.76, "temp_min": 274.99, "temp_max": 277.22, "pressure": 1009, "humidity": 84 },
"visibility": 10000, "wind": { "speed": 1.34, "deg": 252, "gust": 2.68 }, "clouds": { "a
ll": 92 }, "dt": 1649540140, "sys": { "type": 2, "id": 2020678, "country": "PL", "sunris
e": 1649477001, "sunset": 1649525235 }, "timezone": 7200, "id": 3096472, "name": "Kat
owice", "cod": 200 }
```

A następnie dane te zostają odczytywane i przekształcane tak, by działały w aplikacji.

5. PODSUMOWANIE

Aplikacja powstała zgodnie z początkowymi założeniami, spełnia wszystkie wyznaczone w projekcie wymagania. Jest to funkcjonalna aplikacja do sprawdzania pogody: dokładnej temperatury, ciśnienia atmosferycznego, prędkości wiatru, wschodu i zachodu słońca. Wykonana zgodnie z zasadami Material Design. Zawiera również widok dla osób starszych z wyraźnie większymi czcionkami. Istnieje również możliwość wyszukiwania miast.

Jeśli chodzi o możliwość potencjalnego rozwoju, to dobrym pomysłem jest wzbogacenie aplikacji o prognozy długoterminowe – na przykład na 7 dni do przodu. Na dzień dzisiejszy jednak program uznany jest za skończony.