



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

## **Dokumentacja projektu**

**Przedmiot:** Programowanie urządzeń mobilnych

**Tytuł projektu:** Aplikacja pogodowa na  
urządzenia mobilne

Prowadzący:  
mgr inż. Michał Wroński

Wykonawca:  
Kamil Kopczacki w64117

## Opis założeń projektu

Założeniem projektu było stworzenie aplikacji pogodowej na urządzenia mobilne. Aplikacja pobiera dane z api dostarczanego przez <https://openweathermap.org/>. Dane będą pobierane oraz wyświetlane w przejrzysty i intuicyjny dla użytkownika sposób.

## Specyfikacja wymagań

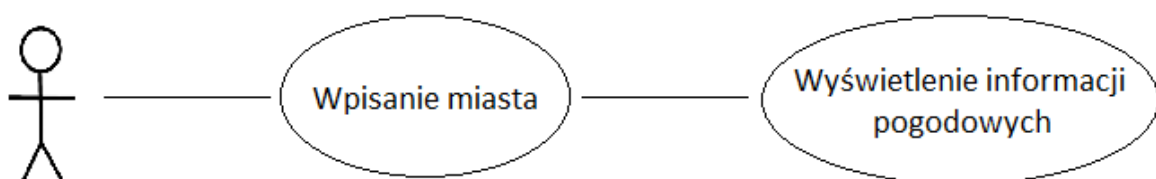
### Wymagania funkcjonalne

- Użytkownik może wpisać oraz wyszukać informacje o pogodzie z wykorzystaniem zewnętrznego serwisu
- Użytkownik powinien zobaczyć pobrane informacje wyświetlone w ładny sposób na ekranie urządzenia

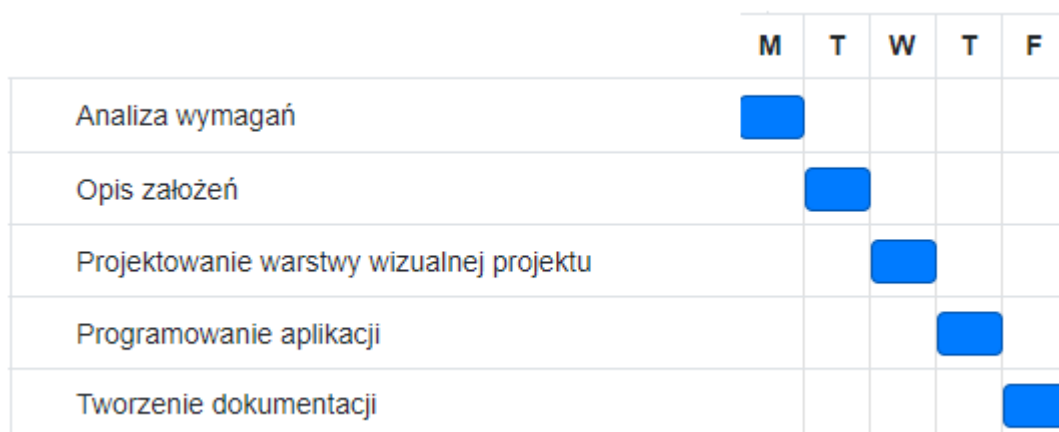
### Wymagania niefunkcjonalne

- Aplikacja jest prosta i intuicyjna w użyciu
- Aplikacja wykorzystuje połączenie sieciowe
- Aplikacja stworzona w technologii Flutter w Android Studio

## Diagram przypadków użycia



## Harmonogram realizacji projektu (diagram Gantt)



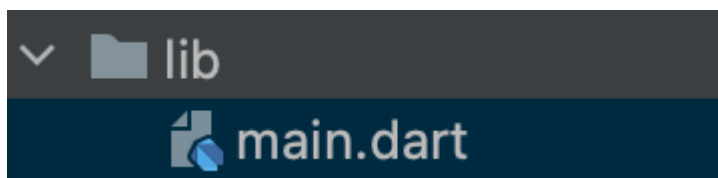
## Opis techniczny projektu

Program został napisany w języku Dart wraz z wykorzystaniem technologii Flutter.

Wykorzystane narzędzia to:

- Dart
- Flutter
- Android Studio
- Emulator iOS

Cała aplikacja ze względu na małą wielkość została napisana w jednym pliku main.dart



Funkcja do pobierania danych z zewnętrznego serwisu wykorzystująca obecnie wpisaną wartość z pola tekstowego oraz klucza do serwisu:

```
40 Future fetchWeather() async {  
41   final inputValue = _inputController.text;  
42   final response = await http.get(Uri.parse('http://api.openweathermap.org/data/2.5/weather?q=$inputValue&appid=$_apiKey'));  
43   setState(() {  
44     data = jsonDecode(response.body);  
45   });  
46 }
```

Funkcja do generowania elementów na ekranie urządzenia:

```
67 generateListItems() {
68   final convertedTemp = (data['main']['temp'] - 273.15 as double).round();
69   final windSpeed = data['wind']['speed'];
70   final humidity = data['main']['humidity'];
71   final pressure = data['main']['pressure'];
72   final sunriseTimestamp = DateTime.fromMillisecondsSinceEpoch(data['sys']['sunrise'] * 1000);
73   final sunrise = sunriseTimestamp.hour.toString() + ":" + sunriseTimestamp.minute.toString();
74   final sunsetTimestamp = DateTime.fromMillisecondsSinceEpoch(data['sys']['sunset'] * 1000);
75   final sunset = sunsetTimestamp.hour.toString() + ":" + sunsetTimestamp.minute.toString();
76
77   return [
78     "Temperatura: $convertedTemp°C",
79     "Prędkość wiatru: $windSpeed km/h",
80     "Wilgotność: $humidity%",
81     "Wschód słońca: $sunrise",
82     "Zachód słońca: $sunset",
83     "Ciśnienie: $pressure hpa"
84   ].map((ListItem) {
85     return Container(
86       child: Row(
87         children: [
88           Text(
89             listItem,
90             style: TextStyle(fontSize: 16),
91           ) // Text
92         ],
93       ), // Row
94     ); // Container
95   }).toList();
96 }
```

Wyświetlenie albo błędu albo danych pogodowych:

```
143 Center(
144   child: (data != null)
145     ? data['cod'] != 200
146       ? Text(
147         "Miejsce, którego szukasz nie istnieje",
148         style: TextStyle(color: Colors.red, fontWeight: FontWeight.bold),
149       ) // Text
150       : Column(
151         children: [
152           ..generateListItems(),
153           Container(
154             height: 100,
155             width: 100,
156             margin: EdgeInsets.only(top: 40),
157             child: getWeatherImage(),
158           ), // Container
159         ],
160       ) // Column
161     : SizedBox.shrink(),
162 ) // Center
```

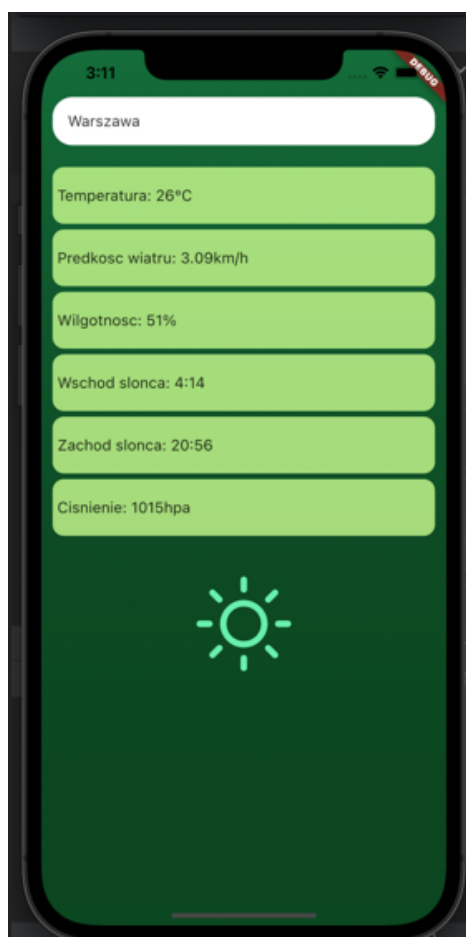
## Prezentacja warstwy użytkowej

Aplikacja stworzona jest tak aby była prosta i intuicyjna w użyciu

Ekran główny aplikacji:



Ekran główny po poprawnym znalezieniu informacji pogodowych danego miasta:



Ekran główny przy wpisaniu błędnego miasta do pola tekstowego:



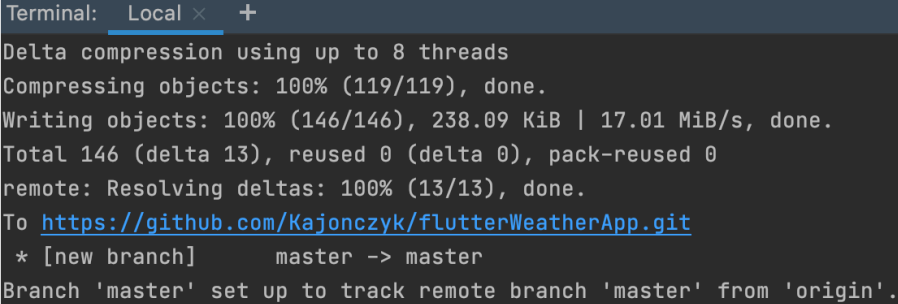
## Raport z testów jednostkowych



## System kontroli wersji

Projekt realizowany był z wykorzystaniem systemu kontroli wersji Git, a wszystkie pliki źródłowe projektu znajdują się pod adresem:

<https://github.com/Kajonczyk/flutterWeatherApp>. Poniżej zdjęcie przesłania plików z wykorzystaniem gita do repozytorium:

A screenshot of a terminal window with a dark background. The title bar at the top says "Terminal: Local x +". The output text is as follows:

```
Delta compression using up to 8 threads
Compressing objects: 100% (119/119), done.
Writing objects: 100% (146/146), 238.09 KiB | 17.01 MiB/s, done.
Total 146 (delta 13), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (13/13), done.
To https://github.com/Kajonczyk/flutterWeatherApp.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

## Dokumentacja projektu

Kod źródłowy wraz z dokumentacją dostępny jest w repozytorium na platformie GitHub pod adresem <https://github.com/Kajonczyk/flutterWeatherApp>