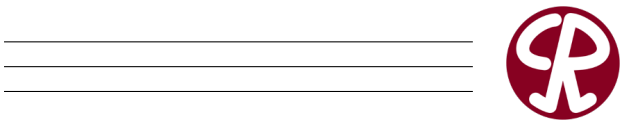
Projekt

Wizualizacja Danych Sensorycznych

MeteoViz - wizualizacja pogody

Emilia Szymańska, 248975



Prowadzący:
dr inż. Bogdan Kreczmer

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki
Politechniki Wrocławskiej

11 maja 2021

Spis treści

1 Charakterystyka tematu projektu

Celem projektu jest stworzenie aplikacji wizualizującej aktualny stan i prognozę pogody dla poszczególnych miejscowości w Polsce. Dla wybranej przez użytkownika lokalizacji będzie odpowiednio zaprezentowana m.in. temperatura, ciśnienie i siła wiatru. Aplikacja zostanie wykonana przy pomocy bibliotek i narzędzi udostępnianych przez Qt skorelowanych z językami QML oraz C++. Kod źródłowy będzie umieszczany na serwisie internetowym wykorzystujący system kontroli wersji Git. Dokumentacja sporządzona zostanie przy pomocy narzędzia Doxygen.

2 Podcele i etapy realizacji projektu

Lista podcelów:

- przegląd zasobów Internetu związanych z tematem projektu, znalezienie źródła danych pogodowych,
- ostateczne zdefiniowanie funkcji dostępnych w aplikacji i zaprojektowanie interfejsu użytkownika,
- nauka podstawowego korzystania z funkcjonalności Qt z wykorzystaniem języka QML oraz C++ (z pomocą materiałów z kursu WDS [2] oraz z samouczków internetowych),
- implementacja podstawowej wersji interfejsu użytkownika,
- zaprogramowanie poboru danych z wybranego serwisu pogodowego,
- aktualizowanie wyświetlanych treści w stosunku do przychodzących danych,
- zaprojektowanie grafik i wzbogacenie nimi interfejsu,
- dopracowanie ewentualnych niedoskonałości aplikacji i stworzenie kompletnej dokumentacji z wykorzystaniem narzędzia Doxygen.

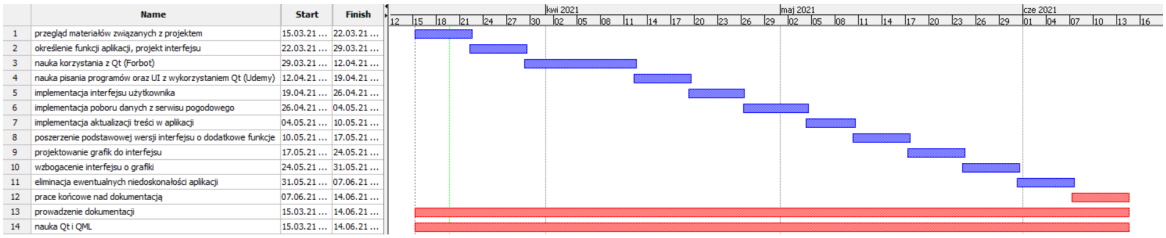
3 Specyfikacja finalnego produktu

Spodziewanym efektem realizowanego projektu jest aplikacja okienkowa, która po uruchomieniu oferuje wybór polskiego miasta, a następnie dla danej lokalizacji zaprezentuje obecny stan pogody oraz prognozy na najbliższe dni. Przy wizualizacji brane będą pod uwagę wartości temperatury $[^oC]$, kierunek i prędkość wiatru $[\frac{m}{s}]$, ciśnienie atmosferyczne $[hPa]$ oraz zależne od dostępności w serwisie wielkość przewidywanych opadów atmosferycznych $[\frac{mm}{h}]$. Całokształt pogody będzie przedstawiony przy pomocy adekwatnej grafiki.

4 Terminarz realizacji poszczególnych podcelów

Poniżej został zamieszczony tygodniowy harmonogram realizacji podcelów (uzupełniony o wykres Gantta (zaprezentowany na rys. 1):

- 22 marca 2021 – zebranie materiałów związanych z tematem (adresy stron z samouczkami, dokumentacją, kursami na np. Udemy itp. zostaną zamieszczone w raporcie), określenie źródła danych pogodowych,
- 29 marca 2021 – określenie funkcji dostępnych w aplikacji, zaprojektowanie interfejsu użytkownika,
- 12 kwietnia 2021 – ukończenie kursu z wykorzystywania Qt na platformie Forbot [3] (implementacja przycisków i rozwijanych list z QtCreator, przesyłanie danych pomiędzy C++ a QML),
- 19 kwietnia 2021 – ukończenie kursu z wykorzystywania Qt na platformie Udemy [1] (wykorzystywanie gniazd i sygnałów, tworzenie obrazków z figur geometrycznych do prezentowania pogody),
- 26 kwietnia 2021 – implementacja podstawowej wersji interfejsu użytkownika (wyświetlanie temperatury, ciśnienia, siły i kierunki wiatru),
- 4 maja 2021 – zaprogramowanie poboru danych z wybranego serwisu pogodowego przez aplikację,
- 10 maja 2021 – odpowiednie aktualizowanie prezentacji graficznej w stosunku do przychodzących danych,
- 17 maja 2021 – rozszerzenie podstawowej wersji interfejsu o dodatkowe funkcjonalności (zależne od danych dostępnych w wybranym serwisie pogodowym),
- 24 maja 2021 – zaprojektowanie grafik urozmaicających interfejs użytkownika w programie graficznym typu Photoshop,
- 31 maja 2021 – wzbogacenie interfejsu użytkownika o zaprojektowane grafiki,
- 7 czerwca 2021 – przetestowanie aplikacji i wyeliminowanie ewentualnych niedoskonałości,
- 14 czerwca 2021 – zakończenie projektu (przedstawienie ostatecznej dokumentacji i efektów końcowych).



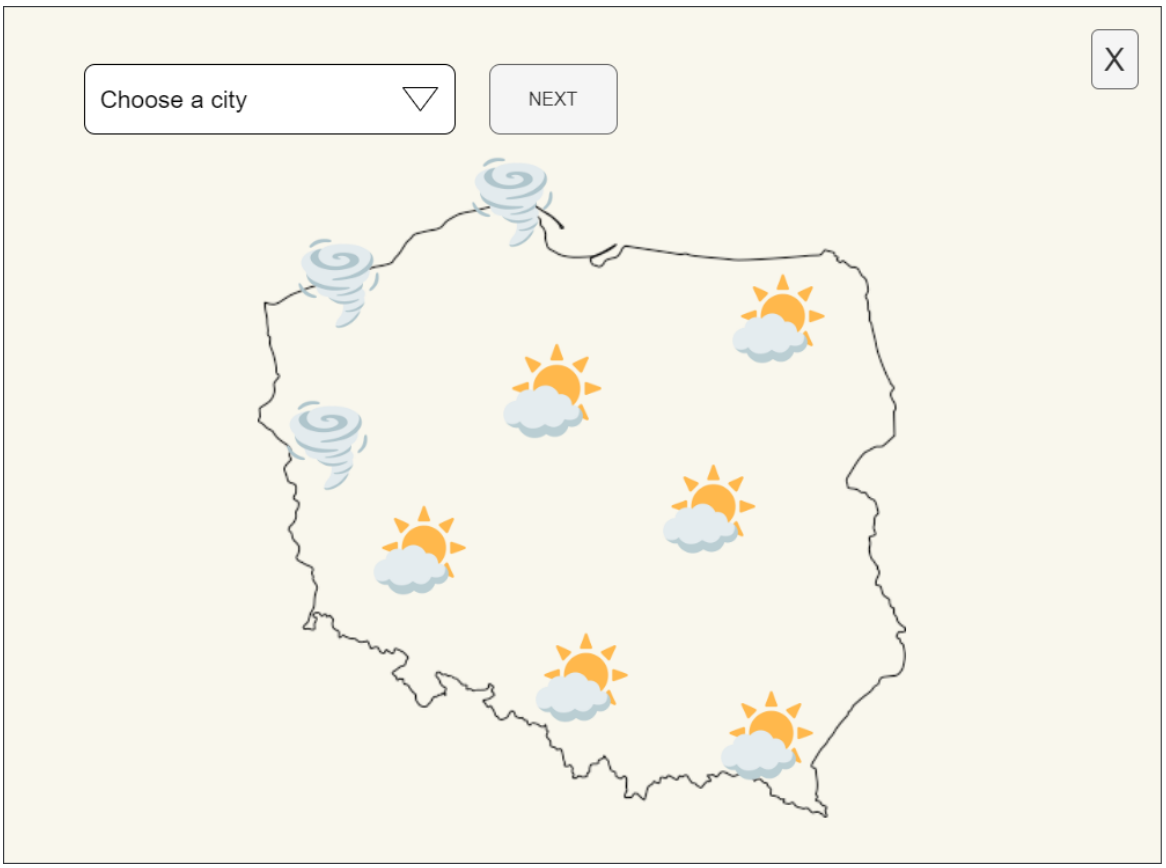
Rysunek 1: Wykres Gantta.

5 Projekt graficznego interfejsu użytkownika

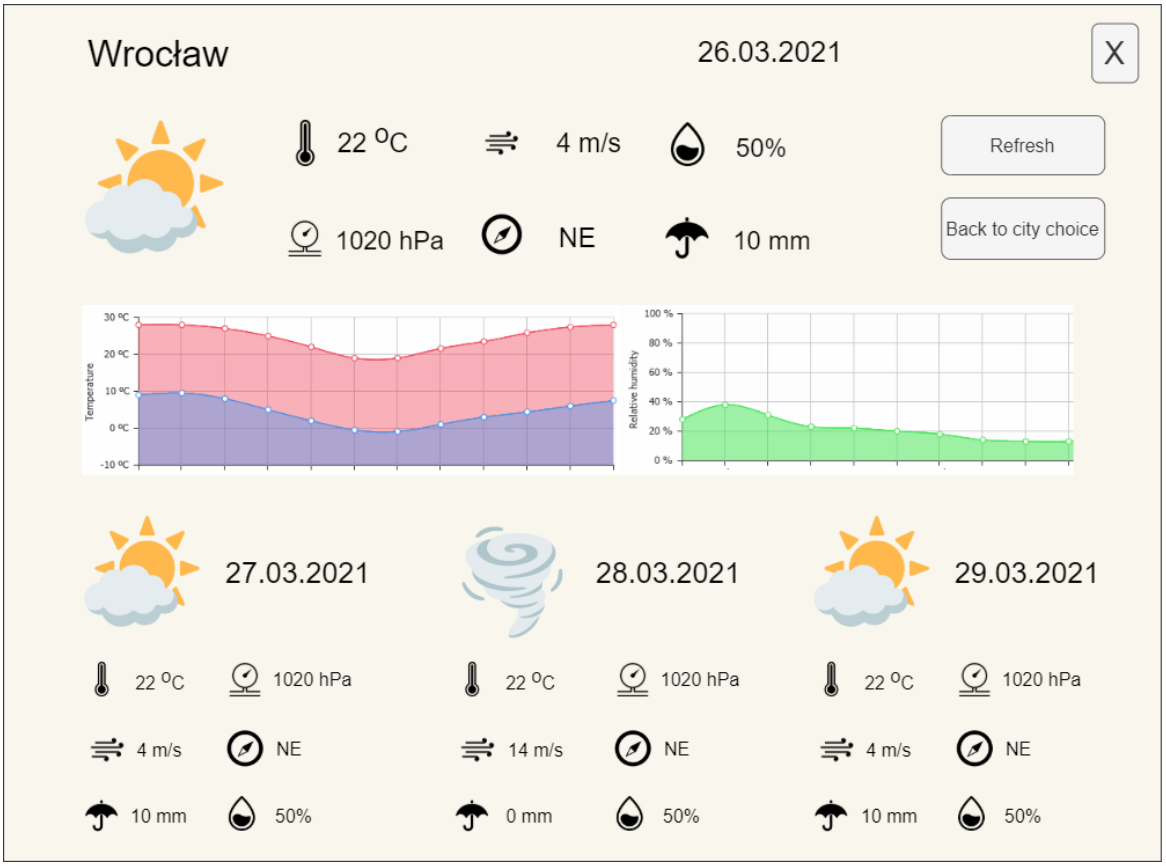
Interfejs składać się będzie z dwóch okien. Pierwsze z nich (rys. 2) umożliwia wgląd w obecny stan pogodowy w kraju (zaprezentowany przez odpowiednie ikony naniesione na mapę Polski) oraz wybór miasta, którego pogodę i prognozy użytkownik chce głębiej poznać.

Drugie okno (rys. 3) zawiera graficzne przedstawienie obecnego stanu pogody oraz trzydniowych prognoz dla wybranego miasta, wartości temperatury powietrza, ciśnienia atmosferycznego, wilgotności, opadów atmosferycznych oraz kierunku i prędkości wiatru. Znajdują się tam również wykresy temperatury i opadów od czasu dla najbliższych godzin.

Użytkownik po uruchomieniu aplikacji wybiera z rozwijanej listy miasto, które go interesuje, co powoduje odblokowanie się przycisku *NEXT*. Po wciśnięciu tego przycisku użytkownik jest przeniesiony do drugiego okna dostarczającego dokładniejsze informacje pogodowe. Możliwe jest odświeżenie danych poprzez wciśnięcie przycisku *REFRESH* (co spowoduje pobranie danych z serwisu i zaktualizowanie zawartości okna) lub powrót do okna z wyborem miasta.



Rysunek 2: Wybór miasta.



Rysunek 3: Stan obecny pogody i prognozy.

6 Rezultaty

6.1 Pobieranie danych

Wybraną stroną internetową z deweloperskim API jest `https://www.tomorrow.io/weather-api/`. Dokumentacja związana z wykorzystywaniem API znajduje się w [6]. Otrzymanie danych sprowadza się do stworzenia adresu URL w odpowiednim formacie, a następnie wysłanie z nim zapytania typu GET do serwera. Podstawowy element tego adresu to *https://data.climacell.co/v4/timelines?* (niedawno firma Climacell została wykupiona przez Tomorrow.io, adres pewnie ulegnie zmianie). Następnie dodaje się oddzielone znakiem '&' dodatkowe dane. Przykładowo, dodanie przyrostka *location=51.10,17.03&fields=temperature&timesteps=current* sprawi, że dla miejscowości znajdującej się na współrzędnych geograficznych [51.10,17.03] zostanie zwrócona obecna wartość temperatury. Można definiować ramy czasowe pobierania danych z określoną częstotliwością (np. od 17:00 26.04.2021 do 06:00 27.04.2021 co godzinę), inne pola (opady, wilgotność itp.), system miar (metryczny/imperialny) oraz strefę czasową, według której podajemy ramy czasowe. Dodatkowo, należy pamiętać o dodaniu pola *apikey=XXX*, gdzie pod XXX trzeba podać wartość własnego klucza wygenerowanego po zarejestrowaniu się na Tomorrow.io. Mój klucz znajduje się w pliku *apikey.txt* dodanym do .gitignore i przy każdej potrzebie wywołania zapytania URL z pliku czytany jest klucz. Do konta w zależności od wybranego pakietu jest ograniczenie liczby wywołań w ciągu godziny. Otrzymane dane są zapisane w formacie JSON, dlatego konieczne jest ich przekonwertowanie.

We fragmencie przedstawionym w sekcji 1 znajduje się odpowiedź serwera na zapytanie o obecną pogodę (*timestep:current*) dla danej lokalizacji. Zwracany jest kod odpowiadający danemu typowi pogody (*weatherCode:1000*).

Listing 1: Format przykładowych danych otrzymywanych z serwera w formacie JSON.

```
{"data":{
    "timelines":[{
        "timestep":"current",
        "startTime":"2021-04-08T22:32:00Z",
        "endTime":"2021-04-08T22:32:00Z",
        "intervals":[{
            "startTime":"2021-04-08T22:32:00Z",
            "values":{
                "weatherCode":1000}
        }]
    }]
}}
```

6.2 Część użytkowa

Część graficzna głównie jest zaimplementowana z wykorzystaniem QML [4], obsługa zdarzeń i logika aplikacji jest opisana za pomocą języka C++ z biblioteką Qt [5].

Pierwsze okno umożliwia rozwinięcie listy z nazwami miast (rys. 4) lub wyszukanie po nazwie (rys.5) i wybranie jednego z nich. Dopiero po wybraniu jakiegoś miasta (rys. 6) możliwe jest wybranie przycisku *NEXT*, które przenosi do drugiego okna. Użytkownik ma też opcję wybrania własnej lokalizacji na mapie poprzez dwukrotne kliknięcie w danym punkcie. Równolegle, trzy miejscowości mają przypięte do swoich współrzędnych geograficznych grafiki reprezentujące odpowiednie warunki pogodowe. Cały czas dostępny jest przycisk *CLOSE* umożliwiający zamknięcie aplikacji.

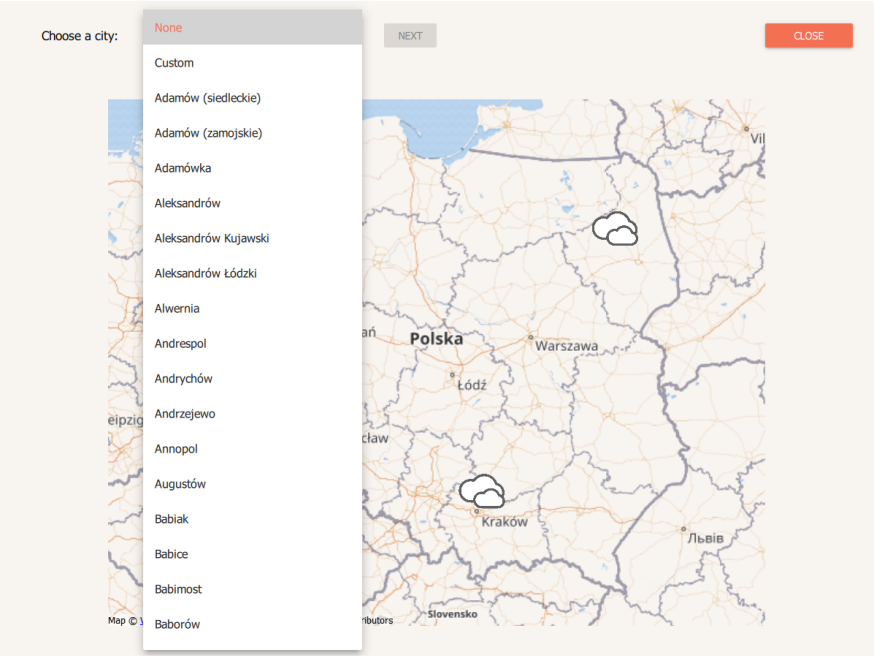
Po wybraniu przycisku *NEXT* użytkownik jest przeniesiony do drugiego okna (rys. 7). Zawiera ono graficzne przedstawienie obecnego stanu pogody oraz trzydniowych prognoz dla wybranego miasta, wartości temperatury powietrza, ciśnienia atmosferycznego, wilgotności, opadów atmosferycznych oraz kierunku i prędkości wiatru. Znajdują się tam również wykresy temperatury i opadów od czasu dla najbliższych godzin. Przycisk *REFRESH* umożliwia zaktualizowanie danych, *BACK* przenosi do poprzedniego okna, a *CLOSE* zamyka aplikację.

7 Kod i dokumentacja

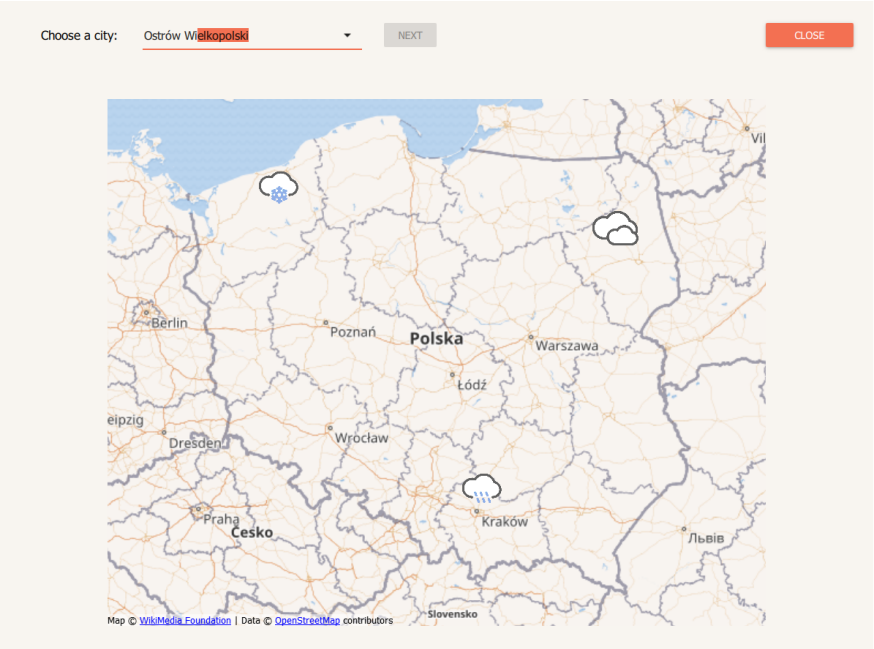
Kod aplikacji można znaleźć w repozytorium pod linkiem:

`https://gitlab.com/emilia-szymanska/meteoviz`.

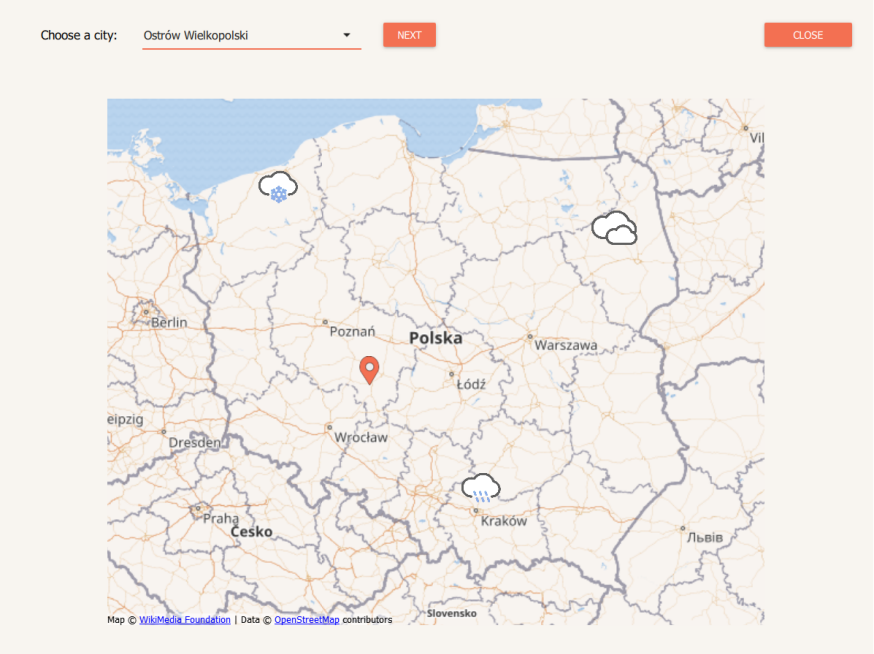
Do każdej klasy dodane są komentarze, dzięki której zostanie wygenerowana dokumentacja Doxygen. Z pomocą narzędzia CI (Continuous Integration) stworzyłam skrypt, który generuje pliki HTML w repozytorium i umieszcza je na stronie internetowej `https://emilia-szymanska.gitlab.io/meteoviz/index.html`.



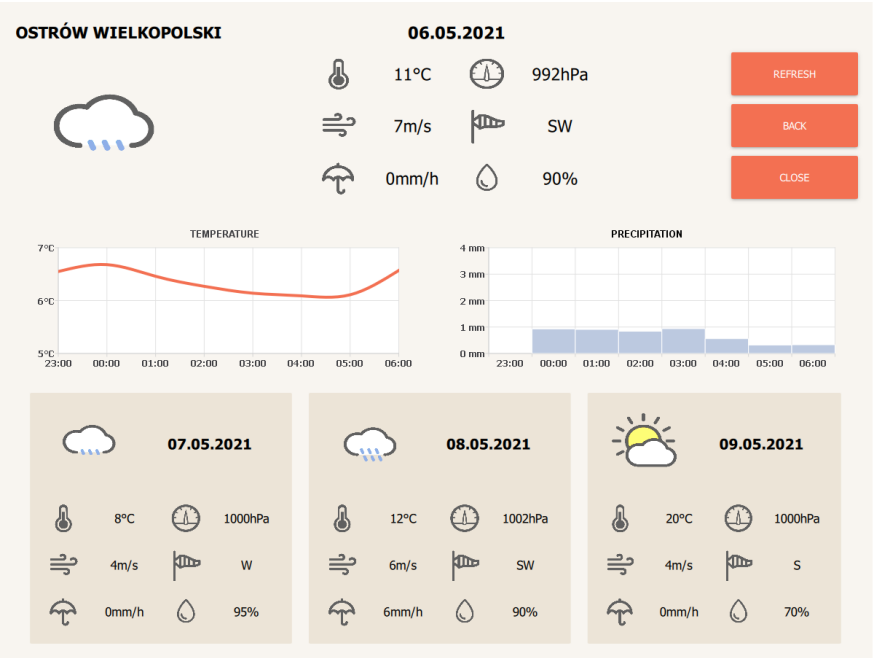
Rysunek 4: Wybór miasta - rozwinięcie listy.



Rysunek 5: Wybór miasta - wyszukiwanie.



Rysunek 6: Wybór miasta po wskazaniu miejscowości.



Rysunek 7: Stan pogody i prognozy.

Literatura

[1] B. Cairns. Qt core for beginners with C++. https://www.udemy.com/course/qt-core-for-beginners/.

[2] B. Kreczmer. Materiały z z wykładów z kursu Wizualizacja Danych Sensorycznych. https://kcir.pwr.edu.pl/ kreczmer/wds/.

[3] M. Patyk. Forbot - kurs programowania w Qt. https://forbot.pl/blog/kurs-qt-1-czym-jest-qt-pierwsza-aplikacja-w-praktyce-id35549.

[4] Qt. QML Qt 5.15 documentation. https://doc.qt.io/qt-5/qmlreference.html.

[5] Qt. Qt 5.15 reference page. https://doc.qt.io/qt-5/reference-overview.html.

[6] Tomorrow.io. TomorrowDocs - weather API reference manual and documentation. https://docs.tomorrow.io/reference/welcome.