

Wizualizacja danych sensorycznych - projekt

1. Temat projektu:

Wizualizacja pogody w Japonii.

2. Wykonawca:

Filip Malinowski 209193

3. Opis projektu:

Projektowany program będzie pobierać pogodę z internetowych serwisów pogodowych. Dane będzie aktualizował dynamicznie co okres czasu wybrany przez użytkownika, np. co 1 minutę, 1 godzinę, 2 godziny. Informacje o pogodzie będą wyświetlane na mapie. Informacje o wybranych miastach będą się wyświetlać na zakładkach z miastami.

4. Opis interfejsu aplikacji:

Interaktywność mapy będzie polegać na tym, że domyślnie dla każdej stolicy prefektury będzie wyświetlana temperatura i ikona zachmurzenia. Po najechaniu kursorem na nazwę miasta rozwinie się szczegółowa pogoda zawierająca: temperaturę, szansę opadów, ciśnienie, wilgotność, zachmurzenie, prędkość i kierunek wiatru oraz poziom promieniowania UV.

W programie będzie również druga zakładka, w której użytkownik będzie mógł wyszukać miasto i wyświetlić dla niego szczegółowe informacje na temat pogody. Wyświetlona zostanie prognoza godzinowa wszystkich czynników podanych wcześniej do wyświetlania na mapie oraz odpowiednie wykresy.

Ma być też możliwość dodania nowych zakładek za pomocą zakładki +. Wtedy jej konfiguracja będzie przebiegać tak samo jak dla drugiej zakładki. Schemat jej zawartości będzie taki sam jak w przypadku drugiej zakładki. Wszystkie zakładki z miastami mają możliwość skasowania się.

5. Źródła pogodowe: Dane pogodowe są pobierane z serwisu developer.forecast.io. Serwis ten pozwala na 1000 darmowych zapytań dziennie ale przy tym udostępniając bardzo dużo danych pogodowych. Dane z tego serwisu pochodzą z: USA NOAA's NEXRAD system, UK Met Office's NIMROD system, The USA NOAA's LAMP system, UK Met Office's Datapoint API, Norwegian Meteorological Institute's meteorological forecast API, USA NOAA's Global Forecast System, USA NOAA's Integrated Surface Database, USA NOAA's Public Alert system, UK Met Office's Severe Weather Warning system, Environment Canada's Canadian Meteorological Center ensemble model, US Navy's Fleet Numerical Meteorology and Oceanography Ensemble Forecast System, USA NOAA and Environment Canada's North American Ensemble Forecast System, USA NOAA's North American Mesoscale Model, USA NOAA's Rapid Refresh Model, Norwegian Meteorological Institute's GRIB file forecast for Central Europe, Norwegian Meteorological Institute's GRIB file forecast for Northern Europe, Worldwide METAR weather reports, USA NOAA/NCEP's

Short-Range Ensemble Forecast, USA NOAA/NCEP's Real-Time Mesoscale Analysis model, USA NOAA/ESRL's Meteorological Assimilation Data Ingest System.

Dane ze wszystkich powyżej wymienionych serwisów są następnie uśredniane i przesyłane do użytkownika na zapytanie wysłane do `developer.forecast.io`. Serwis ten korzysta z bezpiecznego połączenia szyfrowanego protokołem SSLv2.

6. Zrealizowane funkcjonalności:

- Widżet Mapa z mapą Japonii jako tło;
- Obiekt Miasto pobierający i przechowujący dane pogodowe;
- Zakładka przechowująca obiekt klasy Miasto i wyświetlająca wykresy;
- Obiekt wyszukujący koordynaty miasta na podstawie podanej nazwy.
- Widżet ikonka pogodowa wyświetlający ikonę pogodowa
- Widżety wyświetlające wykresy temperaturowe
- Widżety wyświetlające parametr pogody
- Tłumaczenia całego programu do trzech języków
- Dynamiczne dodawanie i usuwanie zakładek z miastami

7. Funkcjonalności do zrealizowania:

- Chmurki pod nazwami miast umieszczonymi na mapie rozwijające się po najechaniu na nie kursorem;
- Belka z ustawieniami dająca możliwość ustawienia okresu odświeżania danych.
- Cykliczne odświeżanie zawartości aplikacji na podstawie ustawionego okresu odświeżania danych.
- Tłumaczenie pop-upów z języka angielskiego na wybrany język z menu dostępnych języków.

8. Opis zaimplementowanych funkcjonalności:

- Widżet Mapa
Przechowuje plik png z mapą Japonii oraz wskaźnik na obiekt klasy QLabel. Wykorzystuje obiekt QLabel w celu ustawienia swojego tła jako mapę Japonii. Rysuje temperatury dla największych miast wszystkich prefektur. Wykorzystuje do tego QPainter.
Docelowo na mapie mają być wyświetlane chmurki przy większych miastach Japonii pokazujące temperaturę i ikonę zachmurzenia w tych miejscach. Po najechaniu na ikonę miasta chmurka ma się rozwinąć przedstawiając szczegółowe informacje nt tego miejsca.
- Obiekt Miasto
Obiekt ten modeluje jedno miasto. Przechowuje koordynaty miasta oraz dane pogodowe na temat tego miasta w formacie json. Łączy się z serwisem internetowym `developer.forecast.io` pobierając z niego dane pogodowe. Następnie konwertuje odebrane dane do formatu obsługiwanego przez klasę składową obiektu Miasto, `JsonObject`. W obiekcie Miasto są metody umożliwiające dostęp do fragmentów klasy `JsonObject` wycinając z niego opis pogody, dane pogodowe, itp.

- **Widżet OknoZZakładkami**
Widżet ten modeluje fragment głównego okna programu będący widżetem z zakładkami. Ma możliwość zmiany swojego rozmiaru wraz ze zmianą rozmiaru głównego okna programu. Tworzy się z dwiema zakładkami: mapą oraz zakładką miasta, która wyświetla informacje pogodowe dla Tokio. Wyświetla zapytanie o nazwę miasta w celu wyświetlenia danych pogodowych. Wyszukane propozycje miast wyświetlane są w ComboBox. Jest jeszcze trzecia zakładka ze znakiem + dodająca kolejne zakładki z miastami. Po dodaniu nowej zakładki z miastem trzeba będzie ustawić miasto tej zakładki. Po tym zostaną pobrane informacje. Skasować będzie można tylko zakładki z miastami. Pierwszej i ostatniej nie da się skasować.
- **Widżet ZakładkaMiasta**
Widżet ten wyświetla aktualną temperaturę, ciśnienie i wilgotność w pierwszym rzędzie. W drugim rzędzie wyświetla wykres zmiany temperatury rzeczywistej oraz odczuwalnej w przeciągu 48 godzin. W trzecim rzędzie wyświetla zakres zmiany temperatury w przeciągu 7 dni.
- **Obiekt WyszukiwarkaMiasta**
Obiekt ten wyszukuje koordynaty miasta na podstawie podanej nazwy. Informacje wyszukuje w serwisie Here firmy Nokia i zwraca odpowiednie trafienia.
- **Widżet IkonkaPogodowa**
Wyświetla ikonkę odpowiadającą aktualnej pogodzie w wyszukiwanym miejscu.
- **Widżet Wykres1**
Wyświetla rozkład temperatury na następne 48 godzin na wykresie oraz wyświetla tekstowy opis prognozowanej pogody na ten czas.
- **Widżet Wykres2**
Wyświetla rozkład temperatury na następne 7 dni na wykresie oraz wyświetla tekstowy opis prognozowanej pogody na ten czas.
- **Widżet DanaPogodowa**
Wyświetla aktualny parametr pogodowy za pomocą QLCDNumber oraz swój tytuł nad liczbą.
- **Dynamiczny tłumacz**
Menu wyboru tłumaczenia programu. Tłumaczy program w locie. Dodatkowe tłumaczenia wystarczy dołączyć do projektu i do zasobów aplikacji, żeby aplikacja automatycznie je wczytała.
- **Dynamiczne dodawanie i usuwanie zakładek**
Zakładki z miastami można dodawać za pomocą przycisku +, oraz usuwać za pomocą przycisków zamykania umieszczonych na zamykalnych zakładkach.

9. Opis funkcjonalności do implementacji:

- Tło w kształcie chmurki pod tekstami wyświetlanymi na zakładce Mapa oraz rozwijające się szczegóły po najechaniu na nazwę miasta.
- Dynamiczne aktualizowanie się danych w wybranym przez użytkownika okresie.

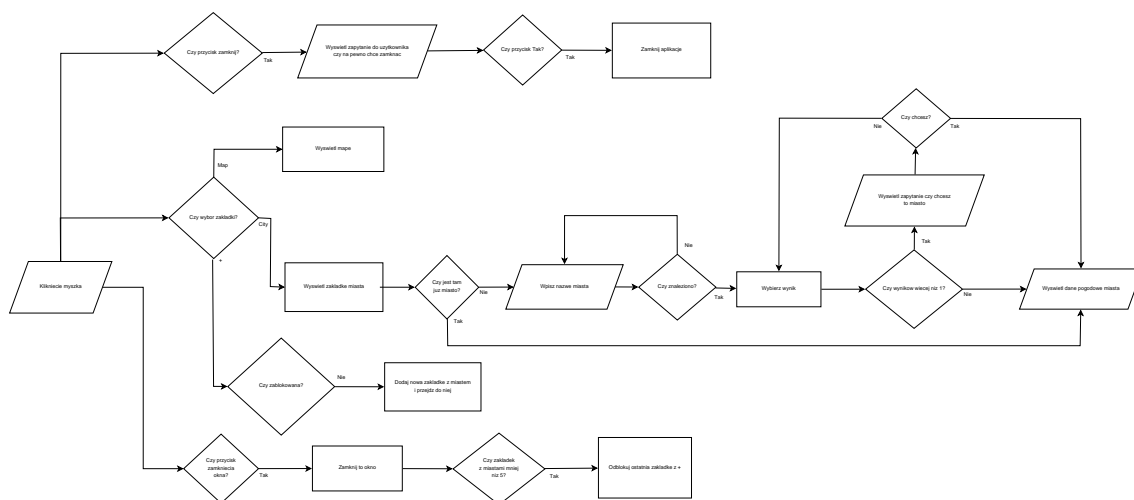
10. Wnioski i podsumowanie: Przy projektowaniu aplikacji najważniejszy jest etap jej planowania. To w tym momencie decyduje się o funkcjonalności aplikacji i od tego momentu zależy czy aplikacja będzie łatwo modyfikowalna i wolna od błędów czy też nie. Na tym etapie ważne jest sprecyzowanie metodyki programowania. Ma to wpływ na późniejsze rozwijanie aplikacji i modyfikowanie jej funkcjonalności. Dokumentacja do projektu jest istotnym elementem pozwalającym osobom trzecim zapoznanie się z projektem i ewentualne wprowadzenie modyfikacji do kodu źródłowego. Bez dokumentacji osoba trzecia nie ma możliwości szybkiego zrozumienia fragmentów kodu co jest potrzebne do pracy w grupie.

Doszedłem do powyższych wniosków na podstawie doświadczenia jakiego nabrałem przy pisaniu tej aplikacji i na podstawie problemów jakie napotkałem. Aplikacja trzyma się przyjętego projektu interfejsu i przy okazji posiada dodatkowe funkcjonalności, które nie były zawarte w projekcie przy etapie projektowania, ale nie kolidowały z interfejsem, ani kodem programu. Części funkcjonalności nie udało mi się zrealizować. Mimo tego kod jest przygotowany na modyfikacje i ewentualne dodanie tychże funkcjonalności.

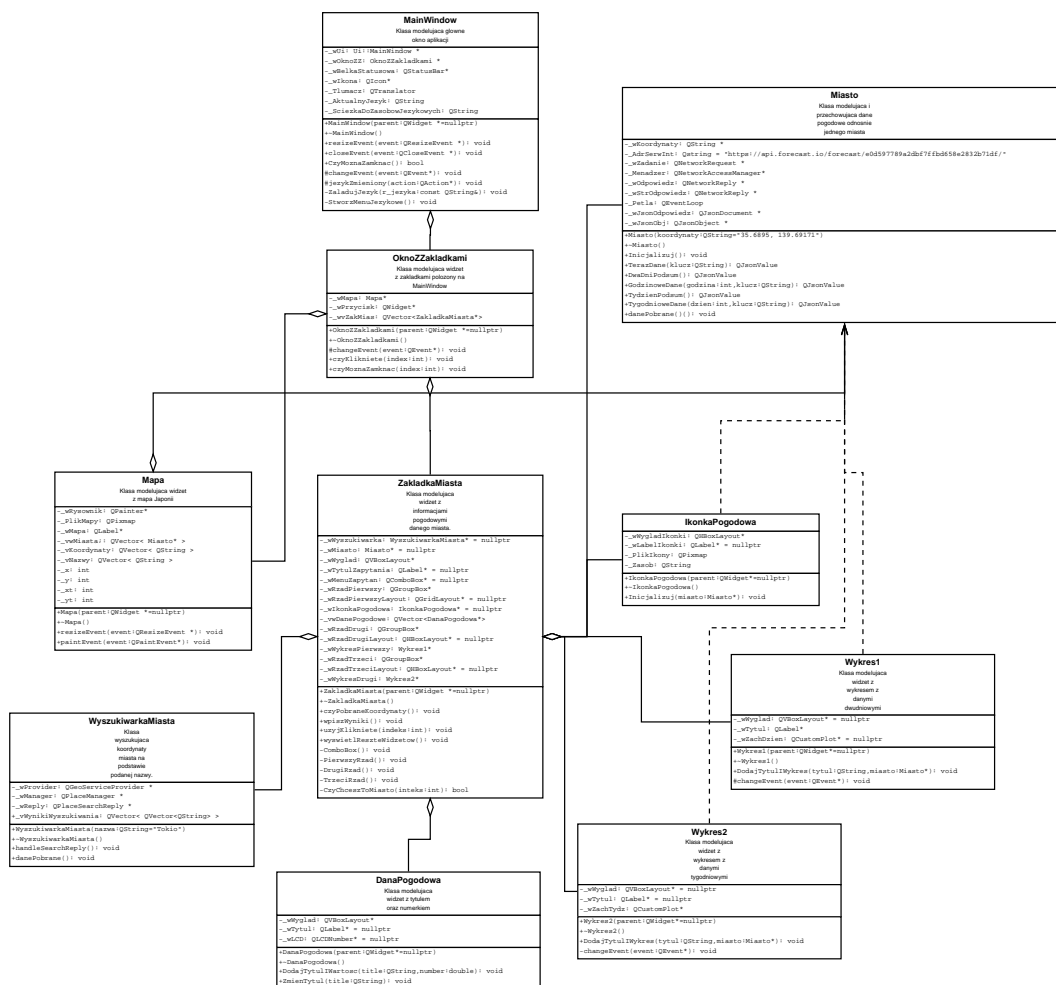
11. Harmonogram:

- 9 kwietnia - zrealizowane** — przeprowadzenie badań na temat najlepszych źródeł informacji oraz odpowiedniej struktury programu;
- 22 kwietnia - zrealizowane** — napisanie wersji alpha programu realizującej podstawowe funkcje programu. Podstawowymi funkcjami do realizacji są: wyświetlenie mapy Japonii, pobranie danych z serwisów pogodowych i zapisanie ich w klasie danego miasta;
- 23 kwietnia - zrealizowane** — podsumowanie wstępnej wersji programu oraz ewentualne poprawki w planach projektu;
- 29 kwietnia - zrealizowane** — przekazanie wstępnej wersji programu do oceny;
- 6 maja - zrealizowane** — dodanie dwóch kolejnych funkcjonalności: zakładki z obiektem Miasto oraz wyszukiwanie koordynatów miasta na podstawie podanej nazwy;
- 10 czerwca - zrealizowane** — oddanie finalnej wersji programu.
- 20 maja - zrealizowane** — połączenie działania wyszukiwarki miasta z obiektem Miasto oraz wyrysowanie temperatur na mapie Japonii w pierwszej zakładce.
- 10 czerwca - zrealizowane** — końcowa wersja projektu do oddania.

12. Diagram przepływu sterowania:



13. Diagram klas:



14. Zarządzanie projektem:

- Git - do przechowywania i rozwijania dokumentacji oraz oprogramowania związanego z projektem. Adres internetowy repozytorium.
- LaTeX - do tworzenia dokumentacji projektu