

# Dokumentacja projektu

Miachail Legczylin Mateusz Nizwantowski Jakub Rymarski

Styczeń 2023

## 1 Cel projektu

Głównym celem tego projektu jest monitorowanie sytuacji w ruchu lotniczym nad Polską w czasie rzeczywistym. Aplikacja jest przyjazna dla użytkownika. Jej założeniem jest łatwa obsługa i przejrzysty sposób prezentowania danych. Użytkownik może korzystać z interfejsu, który w przystępny sposób pozwoli mu wybierać interesujące go dane, takie jak: współrzędne danego samolotu, jego prędkość, czy kraj pochodzenia.

## 2 Określenie grupy docelowej

Ruch lotniczy przeżywa rozkwit w ostatnich latach. Otwarcie się świata na nowe kontakty zagraniczne i rozwój turystyki sprawiają, że wielu ludzi interesuje się podróżami i nowymi technologiami. Obserwują oni chętnie ruch lotniczy. Często właśnie w taki sposób czerpia inspiracje do własnych podróży.

Jednak nasz produkt jest skierowany nie tylko do podróżników, ale także dla ciekawskich którzy chcą wiedzieć jaki samolot właśnie przeleciał nad ich głowami oraz dla ludzi, którzy z racji wieku czy chorób nie mogą już podróżować. Poczują się oni jak pasażerowie samolotu i oderwa się od przyziemnych problemów :).

### **3 Wymagania, które według konspektu ma spełniać aplikacja**

#### **3.1 funkcjonalne**

- czysty kod,
- obszerna dokumentacja,
- pokazywanie prawdziwych informacji

#### **3.2 niefunkcjonalne**

- informatywność,
- prostota obsługi,
- przejrzystość i czytelność UI,
- estetyczny wygląd interfejsu użytkownika,
- wydajność (szybkie i płynne działanie aplikacji),
- niezawodność (brak błędów wpływających negatywnie na użytkowanie aplikacji)

## 4 Struktura projektu

W projekcie wyróżniliśmy konkretne pakiety, które odpowiadają za poszczególne funkcje naszej aplikacji. Uporządkowanie kodu pomaga w zrozumieniu idei projektu, a także ułatwia prace w grupie i potencjalne rozwijanie projektu w przyszłości.

W pakiecie 'constants' przechowujemy: w klasie APIcredentials login użytkownika i hasło do konta na stronie OpenSky, które pozwalają na pobranie danych z tej strony. Nie jest ona przesyłana do publicznego repozytorium dzięki zawarciu jej w pliku gitignore. Aby wiedzieć jak wygląda jego struktura i jak go przygotować stworzona została klasa API\_credentials\_example, która wygląda identycznie nie przechowuje jednak danych wrażliwych. W klasach ActionPerformed i PlainLayout zapisujemy stałe (np. wysokość i szerokość przycisków, czy wysokości i szerokość geograficzna krawędzi mapy), które są wykorzystywane przy rysowaniu mapy i tworzeniu GUI.

W pakiecie 'gui' tworzymy nasz interfejs użytkownika. Pakiet ten zawiera takie klasy jak: DynamicPlacement - w tej klasie odświeżamy dane, ponieważ tak samo jak na żywo tak i w naszej aplikacji sytuacja na niebie zmienia się co kilka sekund), ButtonsFormation - klasa ta odpowiada za przyciski, klasa ComponentPlacement - rozmieszcza dane obiekty na ekranie, ImageWrapper - sprawia, że obrazki i ikonki, przechowywane w innym pakiecie resources, zmieniają pozycję i są ustawiane pod odpowiednim kątem (np. zmiany położenia samolotu), InterMap i MapOrganiser - zarządzają mapą tak, aby dopasowywała się do poleceń użytkownika i wskazywała dokładniej obszary, które go interesują (np. przybliżanie, oddalanie, przesuwanie w górę, w dół, na boki) i wreszcie klasa App - gdzie łączymy naszą aplikację w całość.

Folder resources zawiera wszystkie pliki png, jpg i tym podobne do których odwołujemy się w kodzie. Jest tam między innymi mapa, tekstury używane do przycisków nawigacji po mapie w gui, ikonki samolotów i lotnisk.

Pakiet 'planes' odpowiada za pobieranie danych o samolotach ze strony <https://opensky-network.org/> i przetwarzanie tych informacji. W klasie Airport tworzona jest lista lotnisk, a klasa CountryCodes pozwala nam na przetłumaczenie nazw państw angielskich na polskie. W klasie Plane parsujemy dane otrzymane z API i tworzymy z nich listę samolotów, robiąc to nadajemy naszym danym formę z którą się przyjemnie pracuje i która pozwala na wyświetlenie jej dla użytkownika w prosty i zrozumiały sposób.

W repozytorium znajdują się też pliki z dokumentacją, konspektem i prezentacją oraz pythonowy notebook, który został użyty do tworzenia mapy (w tej części projektu nie użyliśmy javy gdyż uważamy, że każde narzędzie ma swoje zastosowanie, i do takiego zadania najlepiej nadaje się jakiś język skryptowy, w tym przypadku python).

## 5 Jak używać nasz aplikacji?

Nasza aplikacja została zaprogramowana w języku Java w środowisku IntelliJ IDEA. Użytkownik może uruchomić aplikację poprzez zbudowanie projektu i wywołanie metody main w klasie Main poprzez przycisk 'Run'. Zadbaliśmy o to aby od tego momentu wszystko było proste i intuicyjne. Dostępny jest dla użytkownika przyjazny interfejs GUI z kolejnymi elementami:

Interaktywna mapa Polski – użytkownik może poruszać się po niej za pomocą przycisków. (na północ, południe, wschód, zachód). Używając lupy może także przybliżać i oddalać interesujące go obszary. Przeskalowane odpowiednio współrzędne pozwalają nam na uzyskanie proporcjonalnych odległości między danymi samolotami. Dodatkowo na mapie są zaznaczone główne polskie lotniska m.in. w Warszawie, w Krakowie, w Katowicach, we Wrocławiu i w Rzeszowie. Klikając na lotnisko na ekranie wyświetla się podstawowa informacja na temat lotniska.

Ikony samolotów – przedstawiają pozycje danego samolotu. Wybierając dany samolot można wyświetlić informacje o nim - prędkość, kraj właściciela linii lotniczych a także numer kanału porozumiewania się z ziemią. Dodatkowo samoloty są ustawione przodem w kierunku ich lotu. Pozycje samolotów są aktualizowane co około 10 sekund.

Łącząc dane elementy użytkownik dysponuje pomocnym narzędziem do badania i monitorowania ruchu lotniczego nad interesującym go obszarem Polski.