

Szkoła Główna Gospodarstwa Wiejskiego
w Warszawie
Wydział Zastosowań Informatyki i Matematyki

Bartosz Matyjasiak
185117

Projekt i implementacja aplikacji mobilnej
wyświetlającej aktualne lokalizacje
autobusów oraz tramwajów w Warszawie
TODO

Praca dyplomowa inżynierska
na kierunku – Informatyka

Praca wykonana pod kierunkiem
dr. hab. inż. Leszek Chmielewski, prof. SGGW
Instytut Informatyki Technicznej
Katedra Sztucznej Inteligencji

Warszawa, 2020¹

¹Dokument skompilowano z klasą SGGW-thesis w wersji 1.05. Aktualną wersję klasy można pobrać ze strony <http://stud.lchmiel.pl> → Seminarium dyplomowe.

Oświadczenie promotora pracy

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia tej pracy w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis promotora pracy

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej, w tym odpowiedzialności karnej za złożenie fałszywego oświadczenia, oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności z ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. Nr 90 poz. 631 z późn. zm.)

Oświadczam, że przedstawiona praca nie była wcześniej podstawą żadnej procedury związanej z nadaniem dyplomu lub uzyskaniem tytułu zawodowego.

Oświadczam, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną. Przyjmuję do wiadomości, że praca dyplomowa poddana zostanie procedurze antyplagiatowej.

Data

Podpis autora pracy

Streszczenie

TODO

TODO

Słowa kluczowe – TODO, TODO, TODO

Summary

TODO

TODO

Keywords – TODO, TODO, TODO

Spis treści

1	Wstęp	9
1.1	Założenia	9
1.2	Grafiki koncepcyjne	9
2	Implementacja	10
2.1	Publiczne API Warszawy	10
2.2	Komponent GlobalContextProvider	10
2.3	Aktualizacja pozycji pojazdów	11
2.4	Pinezki przystanków	11
2.5	Radar	11
2.6	Ukrycie kluczy API w kodzie	12
3	Podsumowanie i wnioski	13
4	Bibliografia	14

1 Wstęp

W dużych miastach komunikacja miejska jest kluczowym aspektem dla mieszkańców. Niestety duże miasta, w tym Warszawa, boryka się z korkami, wypadkami, robotami drogowymi i innymi przez co autobusy czy tramwaje często nie jeżdżą według rozkładu jazdy. Dlatego dobrą informacją dla podróżującego jest lokalizacja GPS autobusu lub tramwaju. Warszawa udostępnia takie dane w projekcie "Otwarte Dane" jednak są one w postaci nieczytelnej dla przeciętnego człowieka. Rozwiązaniem może być aplikacja mobilna, dzięki której użytkownik będzie widział na mapie kiedy dokładnie przyjedzie autobus lub tramwaj.

1.1 Założenia

Aplikacja powinna:

- Pokazywać aktualne pozycje autobusów i tramwajów na mapie
- Pokazywać pozycje przystanków na mapie
- Udostępniać rozkłady jazdy na każdym z przystanków
- Umożliwiać na dodanie linii autobusowej lub tramwajowej do ulubionych
- Umożliwiać na dodanie przystanku do ulubionych
- Wspierać dwa motywy:
 - Jasny
 - Ciemny

1.2 Grafiki koncepcyjne

2 Implementacja

Do implementacji wybrałem React-Native stworzony przez Facebook. Pozwala on na zrobienie aplikacji na telefony z systemem Android oraz iOS przy pomocy jednego kodu źródłowego napisanego w języku JavaScript XML (w skrócie JSX). Wybrałem także moduł react-native-maps, który jest odpowiedzialny za wyświetlanie mapy Google oraz sterowanie nią.

2.1 Publiczne API Warszawy

Miasto udostępnia dane w postaci publicznego API. Z pośród wielu punktów końcowych tego API są dostępne:

- Pozycje pojazdów danej linii
- Zbiór linii, które odjeżdżają z danego przystanku
- Rozkład jazdy dla danej linii z danego przystanku
- Zbiór wszystkich przystanków

Co ważne pozycje pojazdów są aktualizowane co 10 sekund i też z taką częstotliwością będą aktualizowane w aplikacji. Wszystkie z wymienionych punktów końcowych API zaimplementowałem w klasie `WarsawAPI`.

2.2 Komponent `GlobalContextProvider`

Stworzyłem komponent nadrzędny, przy pomocy kontekstu dostępnego w React, odpowiedzialny za całą logikę aplikacji. Komponent ten przechowuje zmienne:

- Zbiór wszystkich przystanków
- Zbiór ulubionych linii
- Zbiór ulubionych przystanków
- Aktualny wyświetlany region mapy
- Pozycje radaru oraz jego promień
- Zaznaczony przystanek lub pojazd

Oraz funkcje do modyfikacji zmiennych. Komponent też przechowuje referencje do komponentu mapy oraz udostępnia funkcje od sterowania nią.

Jednak nie umieściłem w nim logiki aktualizowania pozycji pojazdów gdyż każda zmiana stanu tego komponentu powoduje przerysowanie wszystkich komponentów `GlobalContext.Consumer`, a wraz nim wszystkich jego dzieci. Przez to, że ten komponent jest używany w wielu miejscach to każda aktualizacja pozycji pojazdów, a ta jest co 10 sekund, powodowałaby przerysowanie całej aplikacji. To wiązałoby się z utratą na szybkości działania aplikacji.

2.3 Aktualizacja pozycji pojazdów

By aktualizacja przebiegała sprawnie wraz z wyświetlaniem logike aktualizacji umieściłem w komponencie `GMap`. Jest to komponent, który jako dziecko posiada tylko komponent `map`. Jest to ważne bo gdy tylko zmieni się stan komponentu `GMap`, a ten będzie się zmieniał co 10 sekund, to wywoła to przerysowanie tylko komponentu `map`. W tym komponencie zaimplementowałem funkcje, która:

1. Dla każdej ulubionej lub wykrytej przez radar linii są pobierane pozycje pojazdów tych linii
2. Jako pojazdy do wyświetlenia są brane pod uwagę tylko te pojazdy, które są z linii ulubionej lub w promieniu radaru oraz czas wysłania sygnału GPS nie jest starszy niż 6 minut

Funkcja ta jest uruchamiana co minute za pomocą funkcji `setTimeout` wbudowanej w język JavaScript

2.4 Pinezki przystanków

2.5 Radar

Głównym celem radaru jest pokazywanie pinezek pojazdów linii z poza ulubionych. Ogranicza on też ilość pinezek do narysowania. Jest to bardzo ważne bo autobusów i tramwajów w Warszawie jest zbyt duża ilość by efektywnie pokazać je wszystkie na raz.

Działanie radaru jest następujące:

1. Użytkownik za pomocą przycisku ustawia pozycje radaru na środku regionu mapy, który jest aktualnie wyświetlany

2. Dla każdego z grup przystanków jest sprawdzane, czy średnia pozycja grupy jest w promieniu radaru
3. Jeśli jest to wszystkie linie z każdego przynkanka danej grupy są dodawane do zbioru linii radaru. Wszystkie elementy tego zbioru są unikalne

2.6 Ukrycie kluczy API w kodzie

3 Podsumowanie i wnioski

4 Bibliografia

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW w tym
w Archiwum Prac Dyplomowych SGGW.

.....
(czytelny podpis autora pracy)

