

# Dokumentacja powdrożeniowa systemu Freeturilo

Mikołaj Ryll, Mikołaj Terzyk

Promotor: dr Paweł Rzążewski

13.01.2021

## Spis treści

<b>1 Historia zmian</b>	<b>2</b>
<b>2 Abstrakt</b>	<b>2</b>
<b>3 Dokumentacja wdrożenia</b>	<b>2</b>
<b>4 Instrukcja instalacji</b>	<b>3</b>
<b>5 Dokumentacja techniczna</b>	<b>4</b>
<b>6 Instrukcja obsługi</b>	<b>5</b>
6.1 Mapa . . . . .	6
6.2 Trasa . . . . .	10
6.3 Historia . . . . .	12
6.4 Panel administratora . . . . .	13

# 1 Historia zmian

Data	Autor	Opis	Wersja
09.01.2022r.	Mikołaj Ryll, Mikołaj Terzyk	Pierwsza wersja dokumentacji	1.0

## 2 Abstrakt

Dokument stanowi dokumentację powdrożeniodową systemu *Freeturilo*. W sekcji 3. spisana została dokumentacja wdrożenia systemu. Rozdział 4. stanowi instrukcję instalacji systemu *Freeturilo*, a 5. jego dokumentację techniczną. W dziale 6. znaleźć można instrukcję obsługi systemowego interfejsu użytkownika, aplikacji mobilnej.

## 3 Dokumentacja wdrożenia

W architekturze systemu *Freeturilo* wyróżnić można cztery główne moduły:

1. bazę danych *PostgreSQL*, przechowującą w szczególności stacje rowerowe i szczegóły tras między nimi,
2. serwer webowy *Freeturilo Web API*, udostępniający systemowe API, a także odpowiadający za większość logiki systemu,
3. serwis *NextBike Service*, aktualizujący dane dotyczące stacji rowerowych na podstawie informacji pobieranych w czasie rzeczywistym z serwerów *NextBike*,
4. aplikację mobilną *Freeturilo* stanowiącą interfejs użytkownika.

Moduły 1-3 zostały wdrożone jako zasoby chmurowe w ramach usług dostawcy *Microsoft Azure*.

**Baza danych** Baza danych umieszczona została w chmurze w ramach *Azure Database for PostgreSQL*, czyli usługi *DBaaS (Database-as-a-Service)* (pochodnej *SaaS*), która udostępnia system *PostgreSQL* w wersji 11. Dobrane przez nasz zespół parametry usługi to 50 GB pamięci masowej oraz procesor jednordzeniowy z dostępem do 2 GB pamięci operacyjnej.

**Serwer webowy** Moduł *Freeturilo Web API* został wdrożony w chmurze z wykorzystaniem usługi *PaaS Azure App Service*, jako aplikacja webowa zaimplementowana z wykorzystaniem framework'u *ASP.NET*. Zasoby przydzielone wdrożonemu serwisowi to 1 GB pamięci masowej oraz procesor jednordzeniowy z dostępem do 1 GB pamięci operacyjnej. Systemowe API zawarte w tym module jest dostępne w sieci publicznej pod adresem <https://freeturilowebapi.azurewebsites.net>.

**Serwis** Wdrożenie *NextBike Service* odbyło się w sposób analogiczny do wdrożenia *Freeturilo Web API*, z takimi samymi parametrami. Jedyną istotną różnicą jest niedostępność serwisu z poziomu sieci publicznej. Jest to moduł, który nie udostępnia API, ani żadnych innych funkcjonalności, które powinny być dostępne dla podmiotów zewnętrznych.

**Aplikacja mobilna** Aplikacja została zaimplementowana w wersji 8. języka *Java* i jest kompatybilna z systemem *Android* o wersji 6.0 (*Marshmallow*) lub wyższej. W przeciwieństwie do powyższych modułów, wdrożona została jako plik *APK* (*Android Package Kit*). Format ten używany jest do dystrybucji i instalacji pakietów systemu operacyjnego *Android*.

## 4 Instrukcja instalacji

Instalacja każdego z modułów systemu odbywa się niezależnie. Moduły, które działają jako zasoby chmurowe (baza danych, serwer webowy, serwis), nie muszą być instalowane przez użytkowników systemu. Zostały one wdrożone przez nasz zespół i udostępnione w sieci publicznej poprzez API. Warunkiem ewentualnej instalacji wspomnianych modułów jest dostęp do aktywnej subskrypcji na platformie *Microsoft Azure* [1].

**Baza danych** Wdrożenie bazy danych odbywa się w dwóch krokach. Pierwszym z nich jest utworzenie pustej bazy danych w ramach usługi chmurowej *Azure Database for PostgreSQL* [2]. W ramach drugiego kroku odtworzyć należy kopię zapasową bazy danych w utworzonej bazie w chmurze. Jest to proces polegający na przesyłaniu danych z maszyny lokalnej do bazy o przydzielonym publicznym adresie IP przy pomocy narzędzia *pg\_restore* [3].

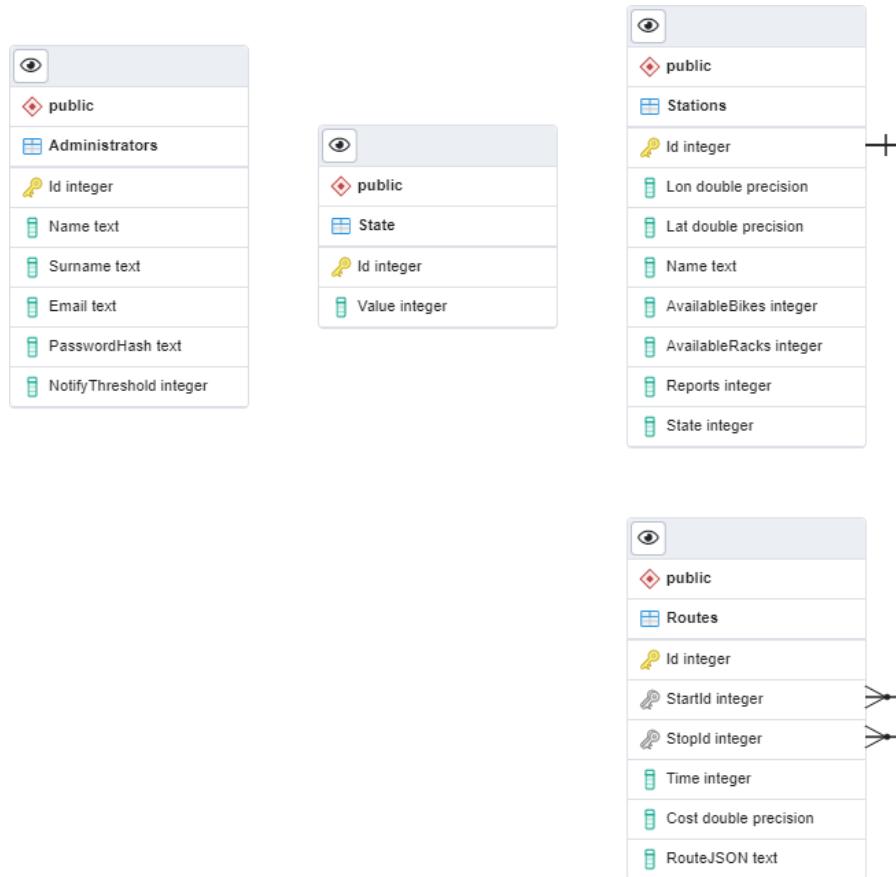
**Serwer webowy i serwis** Kod źródłowy obu modułów jest dostępny w reprezentorium [4]. Należy go pobrać i zbudować projekty *Freeturilo WebApi* oraz *NextBike ApiService* w konfiguracji *Release* [5] [6], po czym opublikować oba na platformie *Microsoft Azure* [7]. Najwygodniejszym do wykonania wymienionych kroków narzędziem jest *Visual Studio 2019*.

**Aplikacja mobilna** Jednym modulem, który musi zostać zainstalowany przez użytkownika w celu korzystania z systemu, jest aplikacja mobilna. Zarówno jej kod źródłowy, jak i plik *APK* dostępne są w reprezentorium [8]. W celu instalacji aplikacji na urządzeniu z systemem *Android*, należy pobrać plik *APK* i umieścić go na urządzeniu, w katalogu dostępnym dla użytkownika (osiągalnym poprzez menadżera plików). Następnie, korzystając z menadżera plików, wybrać należy pobrany plik *APK*. Jeżeli wersja systemu jest niższa niż 8.0, wymagane jest uprzednio zezwolić na instalację aplikacji z nieznanych źródeł w ustawieniach urządzenia. W przeciwnym wypadku, system uprzedzi użytkownika o potencjalnym zagrożeniu. Po zatwierdzeniu nastąpi instalacja pakietu.

## 5 Dokumentacja techniczna

**Serwer Web API** Komunikacja z serwerem polega na wysyłaniu zapytań do wystawionego, publicznego API. Dokładna dokumentacja każdego endpointu API jest dostępna jako część interfejsu i znajduje się pod odpowiednim adresem [9].

**Baza danych** Serwer bazy danych znajduje się pod adresem sieci publicznej [10], a uruchomiony na nim system *PostgreSQL* nasłuchiwa na porcie 5432. Konfiguracja *proxy* zezwala na łączenie się z bazą danych jedynie innym usługom *Microsoft Azure* oraz urządzeniom członków naszego zespołu. Dokładna struktura bazy danych jest przedstawiona w postaci schematu UML (rys. 1).



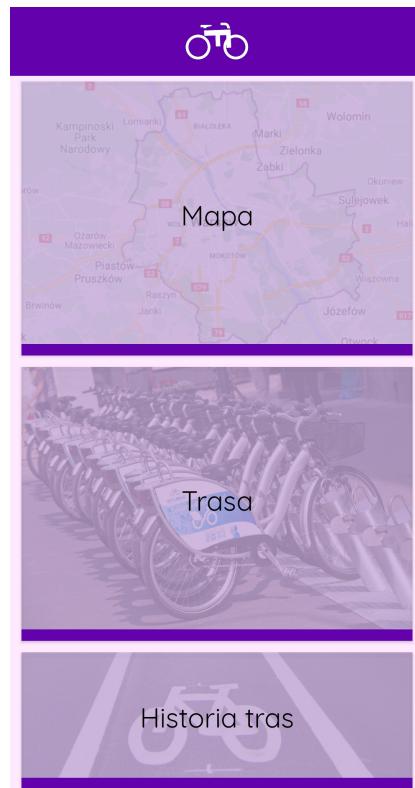
Rysunek 1: Schemat UML bazy danych

**Serwis NextBike i biblioteka do deserializacji** Nie ma możliwości bezpośredniej komunikacji z serwisem przy użyciu urządzeń poza jego wirtualną siecią prywatną. Serwis komunikuje się z serwerem (poprzez API) oraz z interfejsem *NextBike*. Zwrotna z tego interfejsu odpowiedź jest w formacie XML, a jej dokładny schemat znajduje się w repozytorium z dokumentacją [11].

**Aplikacja mobilna** Dokumentacja kodu źródłowego aplikacji mobilnej została wygenerowana przy użyciu narzędzia *javadoc*. Znajduje się ona w całości w katalogu *javadoc* w repozytorium [8]. Aby przeczytać dokumentację, otworzyć należy plik *index.html* przy pomocy przeglądarki internetowej.

## 6 Instrukcja obsługi

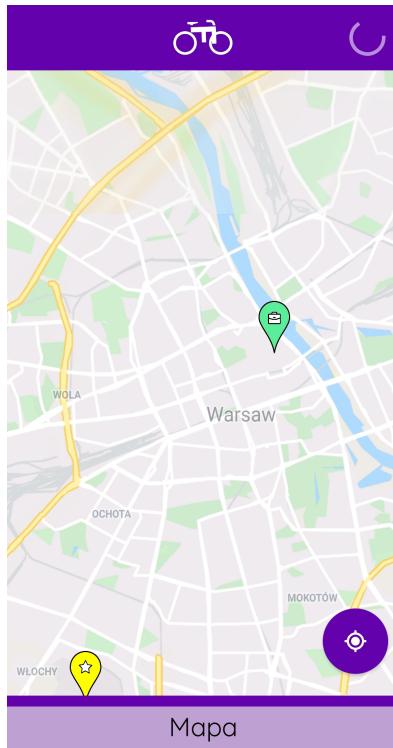
Po uruchomieniu aplikacji wyświetlony zostaje widok **menu głównego** (rys. 2), z którego możliwe jest przejście do widoków mapy, trasy, historii i logowania.



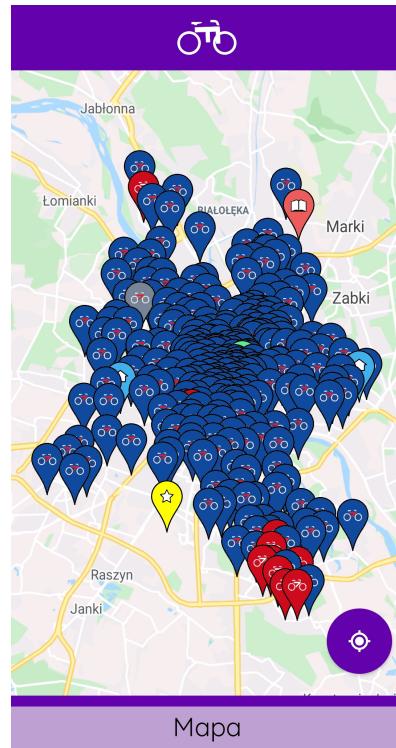
Rysunek 2: Menu główne

## 6.1 Mapa

Aby przejść do widoku **mapy**, należy wybrać przycisk *Mapa* w menu głównym, a następnie zaczekać na pobranie danych o stacjach rowerowych. Trwanie transakcji symbolizuje obracającą się ikoną ładowania w prawym górnym rogu (rys. 3a). Po zakończeniu transakcji danych na mapie ukazane zostają znaczniki wszystkich stacji rowerowych oraz ulubionych miejsc użytkownika (rys. 3b).



(a) Widok mapy podczas ładowania

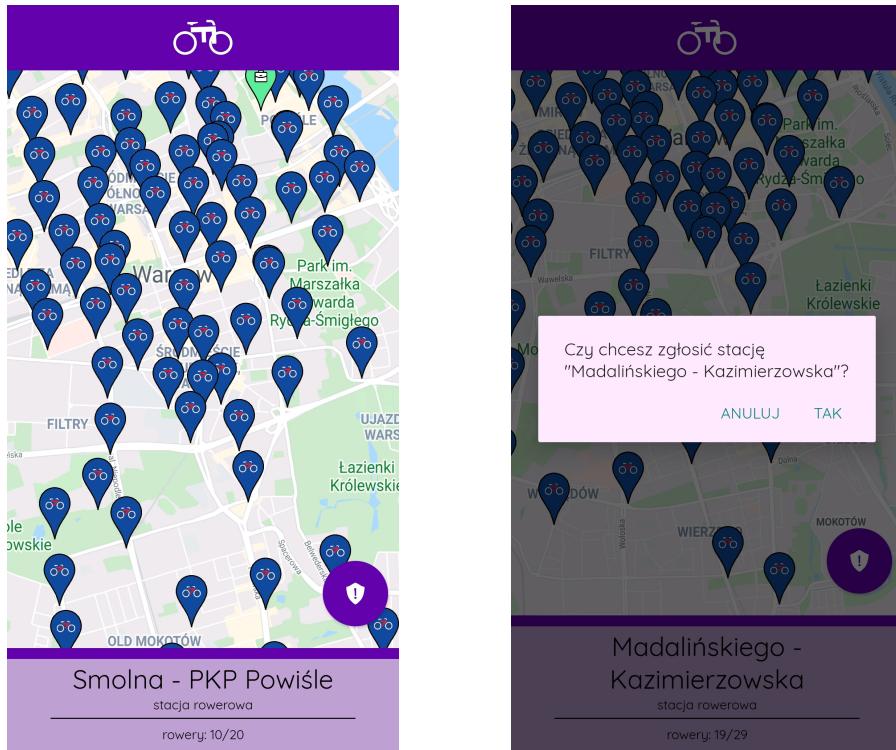


(b) Mapa ze wszystkimi stacjami

Rysunek 3: Widok mapy

Aby dowiedzieć się więcej o **stacji rowerowej** i jej aktualnym stanie wybrać należy jeden ze znaczników z logo *Veturilo*. Szczegóły stacji wyświetlane zostają na dolnym panelu widoku (rys. 4a). Stacje zgłoszone jako niesprawne oraz stacje o małej liczbie rowerów mają na mapie czerwone znaczniki. Stacje niesprawne i stacje puste oznaczone są szarymi znacznikami. Pozostałe znaczniki stacji są granatowe.

Aby **zgłosić** stację jako niesprawną, skorzystać można z wyświetlnego po prawej stronie, zaraz nad dolnym panelem, przycisku, a następnie potwierdzić swój wybór w otwartym oknie dialogowym (rys. 4b).



(a) Szczegóły stacji rowerowej

(b) Zgłoszenie stacji rowerowej

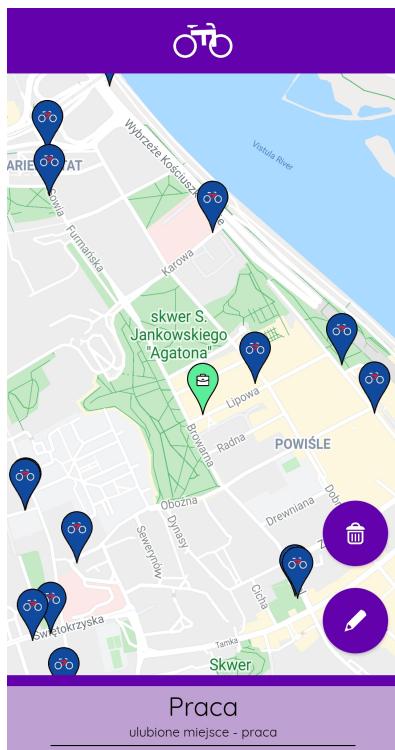
Rysunek 4: Stacja rowerowa

**Ulubione miejsca** użytkownika reprezentowane są na mapie jako znaczniki w jaskrawych kolorach (rys. 5a) z ikonami domu, walizki, książek i gwiazdy, oznaczającymi odpowiednio dom, pracę, szkołę i inne ulubione miejsce.

W celu **dodania** nowego ulubionego miejsca, przytrzymać należy punkt na mapie, a następnie wybierać jego nazwę i typ w otwartym oknie dialogowym (rys. 5b) i zatwierdzić.

Aby **edytować** ulubione miejsce wybrać należy reprezentujący go znacznik na mapie i skorzystać z przycisku z ikoną ołówka w prawym dolnym rogu. Otwarte okno dialogowe może zostać wypełnione podobnie jak w przypadku dodawania nowego ulubionego miejsca.

**Usunięcie** ulubionego miejsca odbywa się poprzez kliknięcie przycisku z koścem umieszczonego nad przyciskiem edycji i zatwierdzenie wyboru.



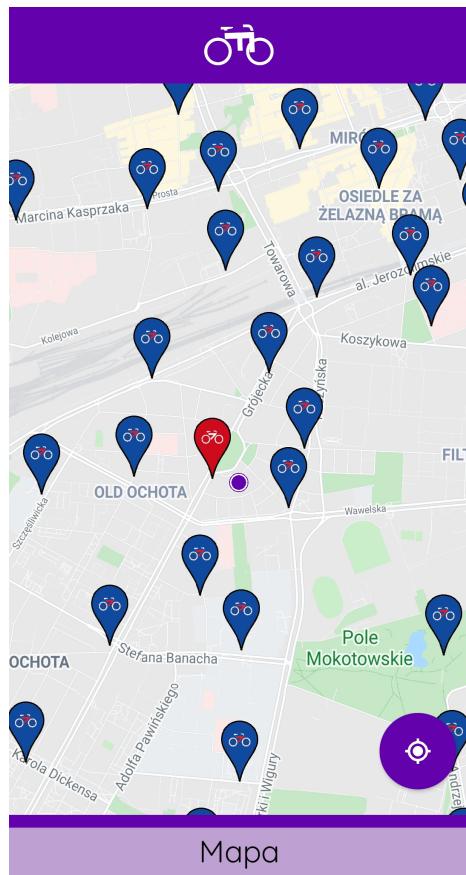
(a) Szczegóły ulubionego miejsca



(b) Dodawanie ulubionego miejsca

Rysunek 5: Ulubione miejsce

Aby zlokalizować swoje urządzenie na mapie, należy włączyć systemowy moduł lokalizacji, a następnie wybrać przycisk wyświetlony w prawym dolnym rogu widoku, gdy żaden ze znaczników nie jest wybrany. Po zezwoleniu na dostęp aplikacji do lokalizacji, na mapie pojawia się fioletowy znacznik (rys. 6).



Rysunek 6: Lokalizacja urządzenia na mapie

## 6.2 Trasa

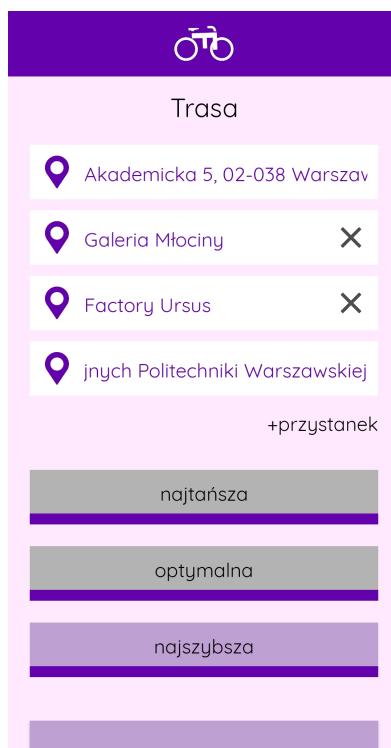
Aby przejść do widoku **wyznaczenia trasy**, skorzystać można z przycisku *Trasa* w menu głównym. Aby wyznaczyć trasę, dobrać należy jej parametry.

Aby dobrać **początkowy** lub **końcowy punkt** trasy należy rozpocząć wpisywanie jego nazwy w odpowiednim polu tekstowym. Po wpisaniu kilku znaków wśród rozwiniętej listy podpowiedzi powinno pojawić się żąданie miejsce. Kolor fioletowy ikony znacznika na początku pola tekstowego oznacza rozpoznanie miejsca przez aplikację (rys. 7a).

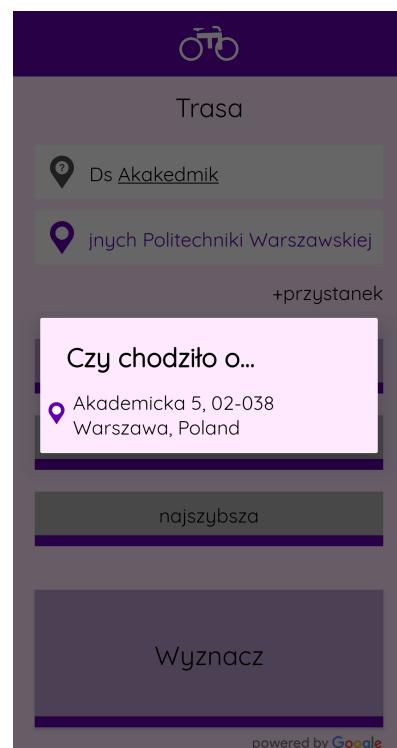
**Dodanie przystanków** do trasy odbywa się poprzez wybór przycisku *+przystanek*. Ukażane pola tekstowe mogą zostać usunięte przyciskiem z krzyżykiem lub wypełnione podobnie jak pola odpowiadające początkowi i końcowi trasy.

**Kryterium wyznaczenia trasy** zostaje wybrane poprzez zaznaczenie jednej z trzech opcji: *najtańsza*, *optymalna* lub *najszybsza*.

Wybór przycisku *Wyznacz* skutkuje wyznaczeniem trasy. Jeżeli jeden z podanych punktów trasy nie został rozpoznany, aplikacja podejmie jeszcze jedną próbę rozpoznania (rys. 7b).



(a) Dobór parametrów trasy

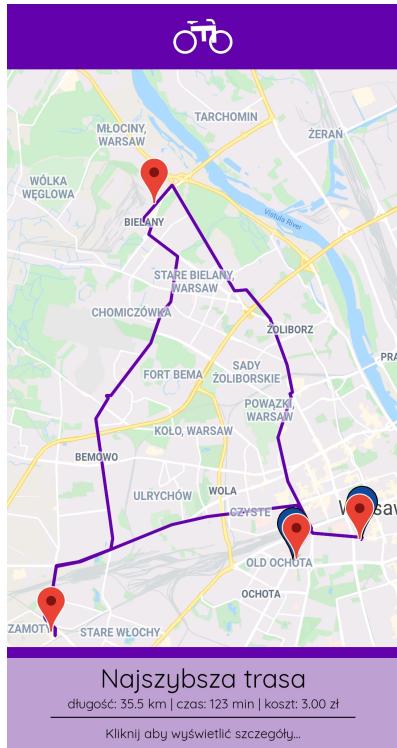


(b) Próba rozpoznania punktu trasy

Rysunek 7: Wyznaczenie trasy

W następującym widoku, po załadowaniu danych, przedstawiona zostaje wyznaczona **trasa**. Punkty na trasie oznaczone są znacznikami o kolorach i ikonach analogicznych do widoku mapy. Czerwonymi, domyślnymi znacznikami oznaczone są punkty, które nie są stacjami rowerowymi ani ulubionymi miejscami użytkownika. Parametry trasy wyświetlane są na dolnym panelu widoku (rys. 8a).

Aby wyświetlić **szczegółowe informacje** o punktach na trasie i przejazdach między nimi należy wybrać dolny panel (rys. 8b). Ponowne kliknięcie dolnego panelu chowa szczegóły trasy.



(a) Mapa z zaznaczoną trasą



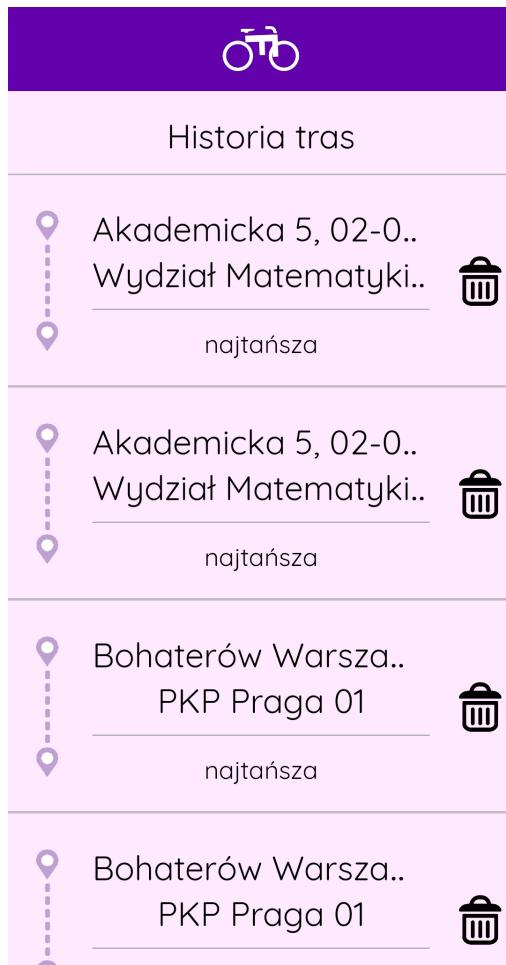
(b) Szczegóły trasy

Rysunek 8: Trasa

### 6.3 Historia

Uruchomienie widoku **historii wyznaczanych tras** odbywa się poprzez wybór przycisku *Historia tras* w menu głównym. Lista wyświetlona w widoku przedstawia parametry wyszukiwanych przez użytkownika tras (rys. 9).

Aby **ponownie wyznaczyć** trasę wybierz odpowiadający jej wpis. Każdy rekord można **usunąć** z historii poprzez przypisany do niego przycisk z ikoną kosza.



Rysunek 9: Widok historii tras

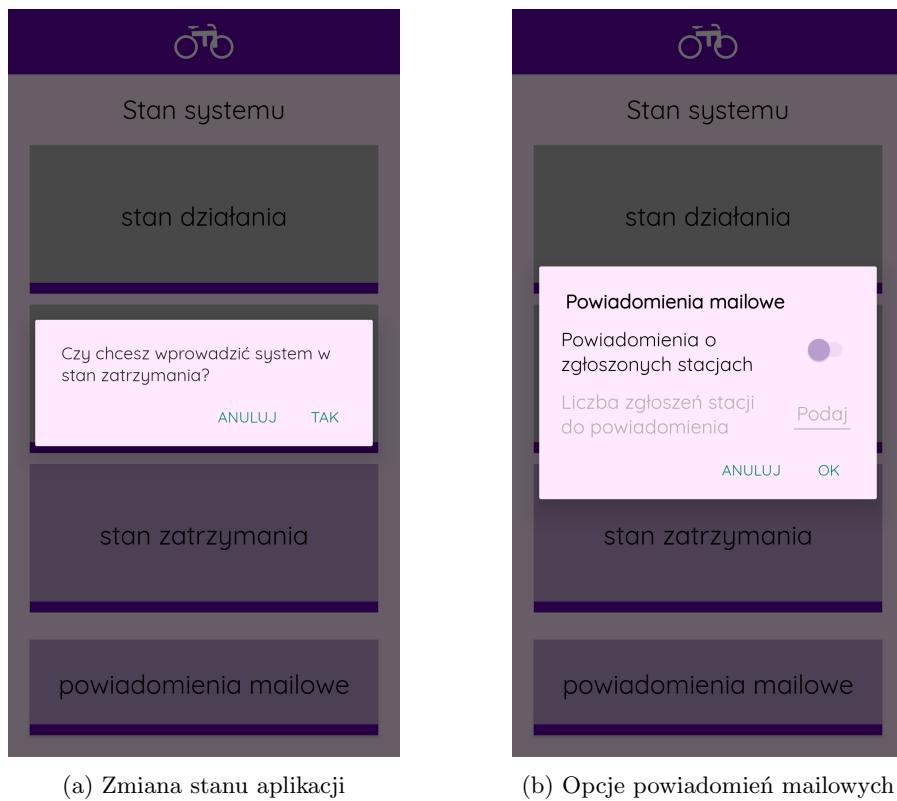
## 6.4 Panel administratora

Aby przejść do widoku logowania, w menu głównym należy nacisnąć i przytrzymać logo aplikacji na górnym pasku. Aby **zalogować się**, wpisać należy email oraz hasło przypisane do swojego konta administratora w odpowiednie pola tekstowe (rys. 10a), a następnie wybrać *Zaloguj*. Po udanym uwierzytelnieniu aplikacja przechodzi do widoku panelu administratora (rys. 10b).



Rysunek 10: Uwierzytelnienie

Aby zmienić stan systemu Freeturilo, należy wybrać jeden z trzech przycisków, a następnie potwierdzić wybór w otwartym oknie dialogowym (rys. 11a). W celu zmiany progu liczby zgłoszeń stacji do powiadomienia mailowego lub wyłączenia powiadomień mailowych, wybrać należy przycisk *powiadomienia mailowe*, dostosować ustawienie w oknie dialogowym (rys. 11b) i zatwierdzić.



Rysunek 11: Funkcjonalności dla administratora

## Literatura

- [1] Dostawca usług chmurowych Microsoft Azure  
<https://azure.microsoft.com/pl-pl/>
- [2] Instrukcja korzystania z usługi Azure Database for PostgreSQL  
<https://docs.microsoft.com/en-us/azure/postgresql/quickstart-create-server-database-portal>
- [3] Narzędzie do przywracania kopii zapasowych baz danych PostgreSQL  
<https://www.postgresql.org/docs/11/app-pgrestore.html>
- [4] Repozytorium z kodem backendu  
<https://github.com/Freeturilo/FreeturiloBackend>
- [5] Budowanie projektu w Visual Studio 2019  
<https://docs.microsoft.com/en-us/visualstudio/ide/building-and-cleaning-projects-and-solutions-in-visual-studio?view=vs-2019>
- [6] Ustawianie konfiguracji Release w Visual Studio 2019  
<https://docs.microsoft.com/en-us/visualstudio/ debugger/how-to-set-debug-and-release-configurations>
- [7] Publikowanie aplikacji ASP.NET w Azure App Service  
<https://docs.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore>
- [8] Repozytorium aplikacji mobilnej Freeturilo  
<https://github.com/Freeturilo/FreeturiloFrontend>
- [9] Dokumentacja interfejsu serwera webowego  
<https://freeturilowebapi.azurewebsites.net/swagger/index.html>
- [10] Adres serwera bazodanowego  
[freeturilo-database.postgres.database.azure.com](https://freeturilo-database.postgres.database.azure.com)
- [11] Schemat XSD  
<https://github.com/Freeturilo/FreeturiloDocumentation/blob/main/nextbike-parser/markers.svg>