

Dokumentacja wstępna

Mikołaj Ryll, Mikołaj Terzyk

28.10.2021

Spis treści

1	Abstrakt	2
1.1	Historia zmian	2
2	Nazewnictwo	2
3	Założenia biznesowe	2
4	Architektura	3
4.1	Freeturilo - aplikacja mobilna	3
4.2	Freeturilo - serwer webowy	3
4.3	NextBike Service	4
5	Wymagania funkcjonalne	4
5.1	Historyjki użytkownika	4
5.2	Przypadki użycia	4
6	Wymagania нефункционалне	7
7	Planowanie	8
8	Podział obowiązków	8
9	Analiza ryzyka	9

1 Abstrakt

Dokument opisuje wstępne założenia projektu *Freeturilo*. Zawiera jego opis ogólny i specyfikację, a w szczególności cele biznesowe, architekturę systemu, wymagania funkcjonalne i niefunkcjonalne, analizę ryzyka, plan pracy i podział pracy nad przedstawianym rozwiązaniem.

1.1 Historia zmian

Data	Autor	Opis	Wersja
17.10.2021r.	Mikołaj Ryll, Mikołaj Terzyk	Pierwsza wersja dokumentacji	1.0

2 Nazewnictwo

Google Maps - platforma stworzona przez Google służąca do planowania tras pieszych, rowerowych, samochodowych, do pobierania zdjęć satelitarnych, interaktywnych panoram i do wielu innych funkcjonalności związanych z geolokalizacją.

Veturilo - warszawski system rowerów miejskich.

NextBike - międzynarodowa firma oferująca rozwiązania uzupełniające komunikację publiczną. Warszawskie *Veturilo* należy do sieci *NextBike*, dzięki czemu możliwe jest pobieranie aktualnych danych o stacjach rowerowych w Warszawie za pośrednictwem interfejsu *NextBike'a*.

Android - najpopularniejszy system operacyjny urządzeń mobilnych.

PostgreSQL - system zarządzania relacyjnymi bazami danych.

Aplikacja mobilna - ogólna nazwa oprogramowania uruchamianego na urządzeniach mobilnych, np. smartfonach czy tabletach.

Serwer webowy - program działający na serwerze internetowym, który obsługuje zapytania protokołu komunikacyjnego HTTP.

Użytkownik - osoba wchodząca w bezpośrednią interakcję z aplikacją mobilną *Freeturilo*.

Administrator - twórca systemu *Freeturilo*, który ma dostęp do większego zakresu funkcjonalności niż użytkownicy.

3 Założenia biznesowe

Warszawski system rowerów miejskich *Veturilo* jest popularną alternatywą dla komunikacji publicznej. Istotną cechą regulaminu wynajmu rowerów jest to, że cena rośnie wraz z czasem wypożyczenia. W szczególności wynajem na czas krótszy niż 20 minut jest bezpłatny.

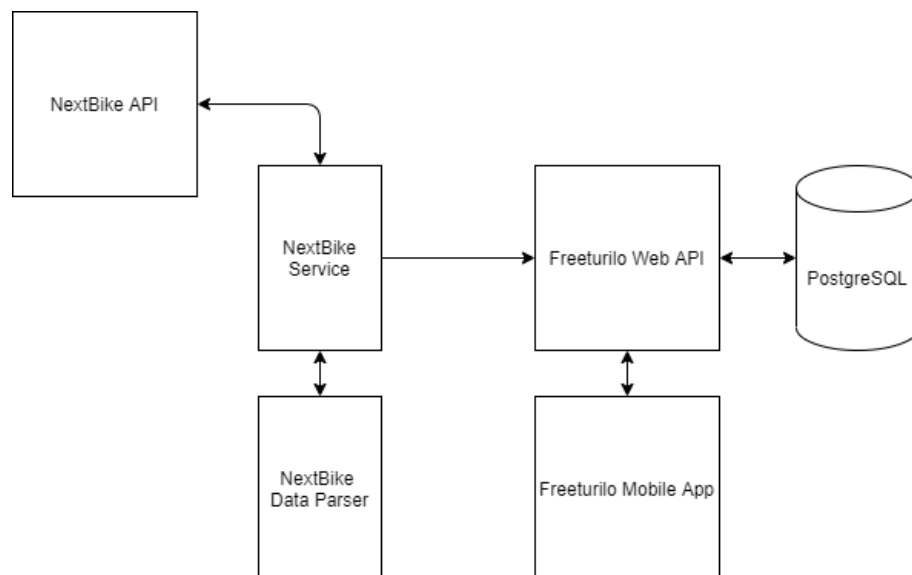
Aplikacja *Freeturilo* to aplikacja mobilna na system Android, która wspomaga planowanie trasy przejazdu rowerami *Veturilo*. System bierze pod uwagę

aktualną dostępność rowerów na poszczególnych stacjach. Ponadto umożliwia wybór trasy według jednego z kryteriów:

- optymalizacja czasu,
- optymalizacja kosztu przejazdu,
- kryterium hybrydowe równoważące dwa powyższe.

Grupą docelową rozwiązania są mieszkańcy Warszawy, w szczególności osoby korzystające z komunikacji miejskiej i rowerzyści. Aplikacja sprawia, że korzystanie z rowerów miejskich przynosi użytkownikom większą oszczędność czasu i pieniędzy.

4 Architektura



Rysunek 1: Schemat architektury systemu

4.1 Freeturilo - aplikacja mobilna

Użytkownik wchodzi w interakcję z systemem używając aplikacji *Freeturilo* na swoim urządzeniu mobilnym. Aplikacja otrzymuje powiadomienia oraz wysyła zapytania do serwera przy pomocy protokołu HTTP.

4.2 Freeturilo - serwer webowy

Serwer udostępnia publiczne API, poprzez które komunikują się z nim pozostałe moduły. Dane przesyłane w odpowiedzi na zapytania są wyznaczane na

podstawie zawartości bazy danych, która przechowuje informacje o bieżącym stanie stacji rowerowych.

4.3 NextBike Service

Oprócz serwera i aplikacji mobilnej uruchomiony jest również serwis *NextBike Service*, który odpytuje API NextBike'a co określoną jednostkę czasu i wysyła nowe dane do serwera, który aktualizuje bazę danych. Serwis korzysta z biblioteki *NextBike Data Parser*, ponieważ udostępnia ona narzędzia do deserializacji odpowiedzi z API NextBike'a, które przesyłane są w formacie XML.

5 Wymagania funkcjonalne

5.1 Historyjki użytkownika

Funkcjonalności aplikacji istotne z punktu widzenia użytkownika przedstawione w postaci historyjek użytkownika.

Jako użytkownik...

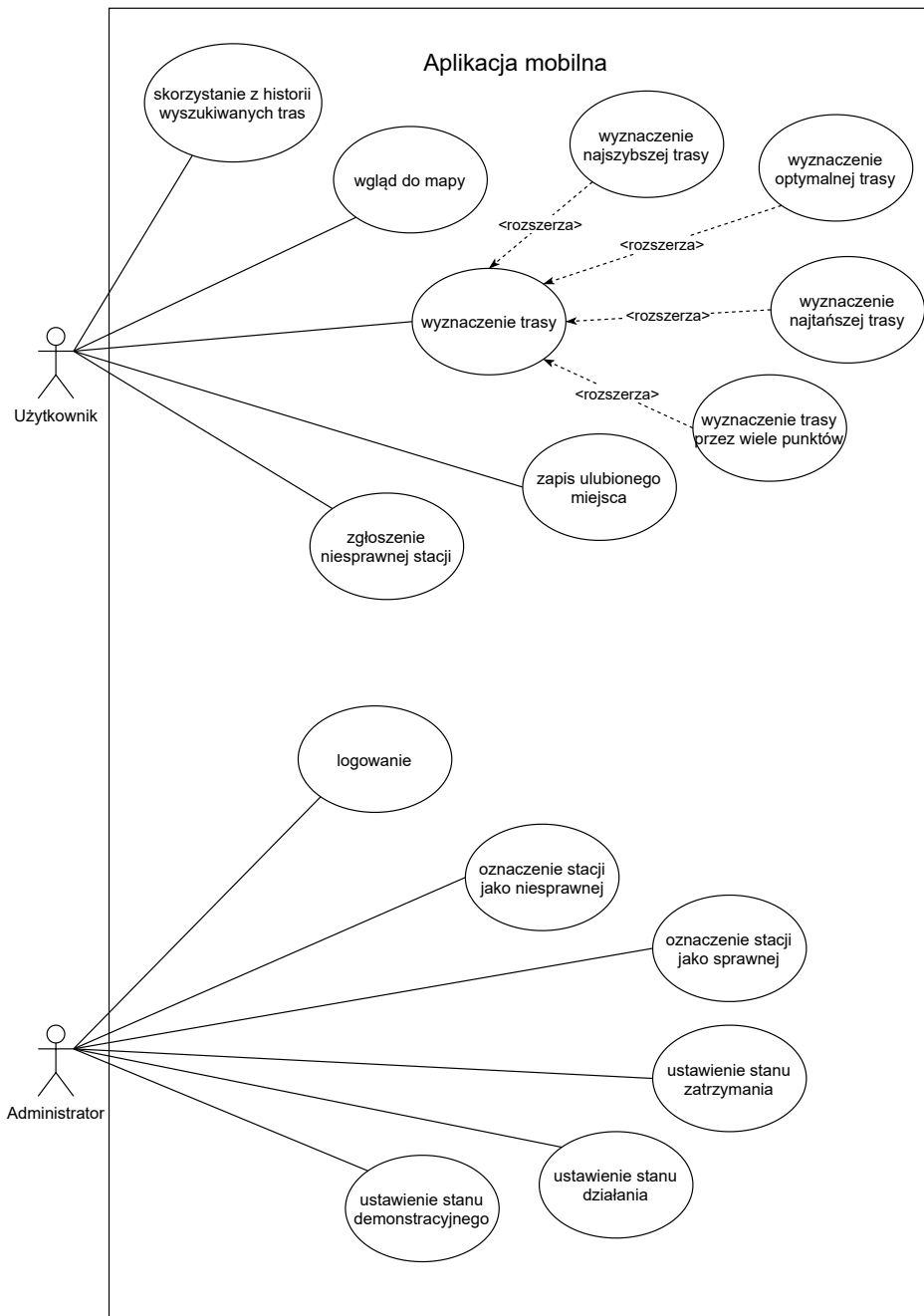
chcę...	aby...
chcę mieć wgląd do mapy	aby przeglądać stacje rowerowe.
chcę wyznaczać najszybszą trasę	aby oszczędzać czas.
chcę wyznaczać najtańszą trasę	aby oszczędzać pieniądze.
chcę wyznaczać optymalną trasę	aby pogodzić obie potrzeby.
chcę wyznaczać trasę przez wiele punktów	aby zatrzymywać się na krótkie przystanki.
chcę zapisywać ulubione miejsca	aby łatwiej wyznaczać do nich trasy.
chcę korzystać z historii wyszukiwanych tras	aby powtarzać wcześniejsze przejazdy.
chcę unikać niesprawnych stacji	aby bez przeszkód dojechać do celu.
chcę zgłaszać niesprawne stacje	aby ułatwić przejazd innym użytkownikom.

Jako administrator...

chcę...	aby...
chcę się logować	aby mieć dostęp do funkcji administratora.
chcę oznaczać stacje jako niesprawne	aby ułatwić przejazd użytkownikom.
chcę usuwać oznaczenia o niesprawności stacji	aby ułatwić przejazd użytkownikom.
chcę wprowadzić system w stan demonstracyjny	aby działał na danych testowych.
chcę wprowadzić system w stan zatrzymania	aby wyłączyć go dla użytkowników.
chcę przywrócić system do stanu działania	aby zapewnić jego pełną funkcjonalność.

5.2 Przypadki użycia

Prezentacja funkcjonalności systemu w postaci diagramu przypadków użycia i tabeli je opisującej.



Rysunek 2: Diagram przypadków użycia

Aktor	Przypadek użycia	Opis	Zachowanie aplikacji
Użytkownik	Wgląd do mapy	Przejdźcie do widoku mapy z zaznaczonymi ulubionymi miejscami i stacjami. rowerowymi	Pobranie danych o stacjach z bazy danych. Pobranie danych o ulubionych miejscach z pliku lokalnego. Wyświetlenie mapy ze znacznikami stacji i ulubionych miejsc.
	Wyznaczenie najszybszej trasy	Wybór punktu startowego i docelowego, wybór kryterium czasu trasy i zatwierdzenie.	Pobranie wyliczonej przez serwer trasy i wyrysowanie jej na mapie wraz z podaniem jej długości, czasu i kosztu przejazdu. Zapisanie trasy w lokalnym pliku historii tras.
	Wyznaczenie najtańszej trasy	Wybór punktu startowego i docelowego, wybór kryterium kosztu trasy i zatwierdzenie.	Pobranie wyliczonej przez serwer trasy i wyrysowanie jej na mapie wraz z podaniem jej długości, czasu i kosztu przejazdu. Zapisanie trasy w lokalnym pliku historii tras.
	Wyznaczenie optymalnej trasy	Wybór punktu startowego i docelowego, wybór kryterium hybrydowego i zatwierdzenie.	Pobranie wyliczonej przez serwer trasy i wyrysowanie jej na mapie wraz z podaniem jej długości, czasu i kosztu przejazdu. Zapisanie trasy w lokalnym pliku historii tras.
	Wyznaczenie trasy przez wiele punktów	Wybór punktu startowego i docelowego, a także kolejnych przystanków na trasie i zatwierdzenie.	Pobranie wyliczonej przez serwer trasy i wyrysowanie jej na mapie wraz ze wszystkimi kolejnymi przystankami. Zapisanie trasy w lokalnym pliku historii tras.
	Zapis ulubionego miejsca	Wybór punktu w widoku mapy, nadanie nazwy, typu (np. dom) i zatwierdzenie.	Zapisanie informacji o ulubionym miejscu w pliku lokalnym.
	Skorzystanie z historii wyszukiwanych tras	Przejdźcie do widoku historii tras i wybór jednej z wypisanych tras.	Pobranie historii tras z pliku lokalnego i wyświetlenie widoku historii tras. Po dokonaniu wyboru ponowne wyznaczenie wybranej trasy.
	Zgłoszenie niesprawnej stacji	Wybór stacji w widoku mapy, i zatwierdzenie zgłoszenia jej niesprawności.	Przesłanie informacji o zgłoszeniu do serwera.

Aktor	Przypadek użycia	Opis	Zachowanie aplikacji
Administrator	Logowanie	Przejsięcie do widoku logowania, wypełnienie pól tekstowych loginem oraz hasłem i zatwierdzenie.	Pobranie informacji o rezultacie próby logowania i ewentualne udostępnienie funkcjonalności administratora.
	Oznaczenie stacji jako niesprawnej	Wybór stacji w widoku mapy, i zatwierdzenie oznaczenia jej jako niesprawnej.	Przesłanie informacji o oznaczeniu do serwera.
	Usunięcie oznaczenia stacji jako niesprawnej	Wybór stacji w widoku mapy, i zatwierdzenie usunięcia jej oznaczenia jako niesprawnej.	Przesłanie informacji o usunięciu oznaczenia do serwera.
	Wprowadzenie systemu w stan demonstracyjny	Wybór stanu demonstracyjnego w widoku zarządzania stanem systemu, i potwierdzenie zmiany stanu.	Przesłanie informacji o zmianie stanu do serwera.
	Wprowadzenie systemu w stan zatrzymania.	Wybór stanu zatrzymania w widoku zarządzania stanem systemu, i potwierdzenie zmiany stanu.	Przesłanie informacji o zmianie stanu do serwera.
	Wprowadzenie systemu w stan działania.	Wybór stanu działania w widoku zarządzania stanem systemu, i potwierdzenie zmiany stanu.	Przesłanie informacji o zmianie stanu do serwera.

6 Wymagania niefunkcjonalne

Wymagania niefunkcjonalne przedstawia poniższa tabela.

Obszar wymagań	Opis wymagania
Użytkowość	Wszystkie funkcjonalności aplikacji są dostępne dla użytkownika na ekranie urządzenia mobilnego o rozdzielczości co najmniej 1280x720 pikseli.
	System operuje na danych nie starszych niż 15 sekund.
Niezawodność	Aplikacja jest dostępna cały czas z przerwami technicznymi, których czas nie przekracza dwóch godzin w tygodniu w godzinach nocnych.
	Moduły systemu są przygotowane na awarię innych modułów i w takim wypadku kontynuują pracę w sposób niezauważalny dla klienta.
Wydajność	Uruchomienie aplikacji wraz z pobraniem danych inicjalizujących nie trwa dłużej niż 5 sekund
	Serwer odpowiada na zapytanie o wyznaczenie trasy nie dłużej niż 3 sekundy.
Utrzymanie	Zachowanie wstecznej kompatybilności w kolejnych wersjach aplikacji oraz serwera.

7 Planowanie

Na projekt zostanie przeznaczone 15 tygodni. Chcemy wykorzystać ten czas jak najefektywniej, dlatego zaplanowaliśmy prace na każdy tydzień. Tabela poniżej zawiera plan pracy.

		Tydzień														
		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
Planowanie	Ustalenie wymagań klienta															
	Przygotowanie wstępnej wersji dokumentacji															
Projektowanie	Wybór architektury i technologii															
	Zaprojektowanie klas, modułów i interfejsów															
Implementacja	Implementacja serwisu aktualizującego bazę danych															
	Implementacja serwera															
	Implementacja aplikacji mobilnej															
Wdrożenie	Przygotowanie testów															
	Napisanie instrukcji															
	Wdrożenie aplikacji na Azure															
	Napisanie dokumentacji powdrożeniowej															
Praca dyplomowa																

Rysunek 3: Plan pracy z podziałem na tygodnie

Cały projekt został podzielony na kilka etapów. W niektórych tygodniach zaplanowany jest więcej niż jeden etap. Będzie to okres, kiedy każdy z nas będzie pracował równolegle nad swoją częścią projektu. O podziale obowiązków więcej w następnym rozdziale.

8 Podział obowiązków

Planowany podział obowiązków prezentuje się następująco:

I) Mikołaj Ryll

1. Przygotowanie dokumentacji rozwiązania
2. Zaprojektowanie logiki aplikacji
3. Implementacja oraz wdrożenie części backendowej
4. Integracja aplikacji z *Google Maps API*

5. Automatyczne testowanie aplikacji
6. Manualne testy działania aplikacji

II) Mikołaj Terzyk

1. Przygotowanie dokumentacji rozwiązania
2. Zaprojektowanie logiki aplikacji
3. Zaprojektowanie interfejsu graficznego
4. Implementacja oraz wdrożenie części frontendowej
5. Automatyczne testowanie interfejsu graficznego
6. Manualne testy działania aplikacji

9 Analiza ryzyka

Przygotowanie aplikacji będzie długotrwałym i złożonym procesem, dlatego istotna jest odpowiednia analiza ryzyka. Poniższa tabela zawiera analizę SWOT projektu *Freeturilo*.

	Zagrożenia	Szanse
Wewn.	<ol style="list-style-type: none"> 1. Brak czasu w związku z innymi projektami 2. Brak odpowiedniej komunikacji z zespołem 	<ol style="list-style-type: none"> 1. Nauka dokładności i terminowości 2. Wzajemna wymiana doświadczenia
Zewn.	<ol style="list-style-type: none"> 1. Brak możliwości testowania w okresie zimowym 2. Zmiana lub zamknięcie API <i>NextBike'a</i> 	<ol style="list-style-type: none"> 1. Udoskonalenie systemu rowerów miejskich 2. Usatysfakcjonowanie klienta oraz zainteresowanie nowych klientów

Realizacja projektu jest związana z zagrożeniami. Największym z nich jest zmiana publicznego interfejsu, który dostarcza informacji o lokalizacji rowerów. Spowodowałaby ona nieprawidłowe działanie aplikacji *Freeturilo* i zmusiłaby nas do natychmiastowej aktualizacji.

Stworzenie aplikacji może również przynieść wiele korzyści. Pozwala przyczynić się do ulepszenia systemu rowerów miejskich oraz zyskać potencjalnych klientów naszego produktu.

Literatura

- [1] Dokumentacja systemu Android
- [2] Dokumentacja API Google Maps
- [3] Aktualna dostępność rowerów