

Dokumentacja wstępna systemu Freeturilo

Mikołaj Ryll, Mikołaj Terzyk

Promotor: dr Paweł Rzążewski

28.10.2021

Spis treści

1	Abstrakt	2
1.1	Historia zmian	2
2	Nazewnictwo	2
3	Założenia biznesowe	3
4	Wymagania funkcjonalne	4
4.1	Historyjki użytkownika	4
4.2	Przypadki użycia	5
4.3	Kluczowe funkcjonalności	8
5	Wymagania niefunkcjonalne	11
6	Planowanie	12
7	Podział obowiązków	12
8	Analiza ryzyka	13
9	Architektura	14
9.1	Freeturilo - aplikacja mobilna	14
9.2	Freeturilo - serwer webowy	14
9.3	NextBike Service	14

1 Abstrakt

Dokument opisuje wstępne założenia projektu *Freeturilo*. Zawiera jego opis ogólny i specyfikację, a w szczególności cele biznesowe, wymagania funkcjonalne i нефункционалне, 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
19.10.2021r.	Mikołaj Ryll, Mikołaj Terzyk	Poprawa schematu architektury, rozszerzenie przypadków użycia o obsługę błędów, dodanie hiperłączy do bibliografii	1.1
21.10.2021r.	Mikołaj Terzyk	Drobne poprawki stylistyczne	1.2

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 nieliniowo 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. Aplikacja sprawia, że korzystanie z rowerów miejskich przynosi użytkownikom większą oszczędność czasu i pieniędzy.

4 Wymagania funkcjonalne

4.1 Historyjki użytkownika

Poniżej, w postaci historyjek użytkownika, przedstawione zostały funkcjonalności aplikacji istotne z jego punktu widzenia.

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 dojeżdżać do celu.
chcę zgłaszać niesprawne stacje	aby ułatwiać przejazd innym użytkownikom.

Jako administrator...

chcę...	aby...
chcę się logować	aby mieć dostęp do funkcji administratora.
chcę oznaczać stacje jako niesprawne	aby ułatwiać przejazd użytkownikom.
chcę usuwać oznaczenia o niesprawności stacji	aby ułatwiać przejazd użytkownikom.
chcę wprowadzać system w stan demonstracyjny	aby działał na danych testowych.
chcę wprowadzać system w stan zatrzymania	aby wyłączać go dla użytkowników.
chcę wprowadzać system w stan działania	aby zapewniać jego pełną funkcjonalność.

4.2 Przypadki użycia

W tej sekcji zaprezentowane zostały funkcjonalności systemu przy użyciu diagramu przypadków użycia (rysunek 1) i tabeli je opisujących.



Rysunek 1: 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 trasy	Wybór punktu startowego i końcowego (opcjonalnie też punktów przystankowych), wybór kryterium i zatwierdzenie.	Pobranie wyliczonej przez serwer trasy i wyrysowanie jej na mapie wraz ze wszystkimi przystankami. Zapisanie trasy w lokalnym pliku historii tras.
	Wyznaczenie najszybszej trasy	Wyznaczenie trasy z wyborem kryterium czasu trasy.	Jak wyżej.
	Wyznaczenie najtańszej trasy	Wyznaczenie trasy z wyborem kryterium kosztu trasy.	Jak wyżej.
	Wyznaczenie optymalnej trasy	Wyznaczenie trasy z wyborem kryterium hybrydowego.	Jak wyżej.
	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	Przejsie 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.
	Oznaczenie stacji jako sprawnej	Wybór stacji w widoku mapy, i zatwierdzenie usunięcia jej oznaczenia jako niesprawnej.	Przesłanie informacji o usunięciu oznaczenia do serwera.
	Ustawienie stanu systemu	Wybór stanu w widoku zarządzania stanem systemu i potwierdzenie ustawienia.	Przesłanie informacji o ustawieniu stanu do serwera.
	Ustawienie stanu demonstracyjnego	Ustawienie stanu systemu na stan demonstracyjny.	Jak wyżej.
	Ustawienie stanu zatrzymania	Ustawienie stanu systemu na stan zatrzymania.	Jak wyżej.
	Ustawienie stanu działania	Ustawienie stanu systemu na stan działania.	Jak wyżej.

4.3 Kluczowe funkcjonalności

Poniżej rozpisana została większa ilość szczegółowych informacji o najistotniejszych mechanizmach działania systemu.

Nazwa	Wgląd do mapy
Opis	Użytkownik może przejrzeć stacje rowerowe i dotyczące ich informacje, np. liczbę rowerów, liczbę stojaków i stan sprawności. Wyświetlane są też ulubione miejsca użytkownika. Z poziomu widoku mapy użytkownik może zgłosić niesprawną stację i zapisać ulubione miejsce.
Warunek wstępny	
Warunek końcowy	Wyświetlona zostaje mapa z naniesionymi znacznikami.
Potencjalne wyjątki	1. Brak połączenia aplikacji z internetem. 2. Niedostępność jednego z niezbędnych komponentów. 3. Niedostępność pliku z ulubionymi miejscami.
Aktorzy	Użytkownik
Wyzwalacz	Użytkownik chce zobaczyć mapę stacji rowerowych.
Standardowa procedura	(1) Użytkownik przechodzi do widoku mapy. (2) Aplikacja wysyła zapytanie do serwera. (3) System przygotowuje dane do zwrócenia. (4) Serwer zwraca dane o stacjach do aplikacji. (5) Aplikacja pobiera dane o ulubionych miejscach z pliku lokalnego. (6) Wyświetlona zostaje mapa z naniesionymi znacznikami.
Alternatywne procedury	(2) Aplikacja nie ma dostępu do internetu. (3) Wyświetlony zostaje widok z informacją o błędzie. (3) Jeden z niezbędnych komponentów jest niedostępny. (4) Serwer zwraca komunikat o błędzie do aplikacji. (5) Wyświetlony zostaje widok z informacją o błędzie. (5) Plik z ulubionymi miejscami jest niedostępny. (6) Wyświetlony zostaje chwilowo komunikat o błędzie. (7) Następuje kontynuacja procedury.

Nazwa	Wyznaczenie trasy
Opis	Użytkownik może wybrać punkt startowy, końcowy i przystanki przy pomocy adresu, mapy lub ulubionych miejsc. Dostępne są kryteria czasu, kosztu i hybrydowe. Trasa zostaje wyrysowana ze wszystkimi przystankami, włącznie z przesiadkowymi stacjami rowerowymi. Wraz z wyznaczeniem trasy system zwraca jej szczegóły: czas, koszt i długość.
Warunek wstępny	
Warunek końcowy	Wyświetlona zostaje mapa z wyrysowaną trasą.
Potencjalne wyjątki	<ol style="list-style-type: none"> 1. Niedostępność pliku z ulubionymi miejscami. 2. Brak połączenia aplikacji z internetem. 3. Niedostępność jednego z niezbędnych komponentów. 3. Niedostępność pliku z historią tras.
Aktorzy	Użytkownik
Wyzwalacz	Użytkownik chce wyznaczyć trasę przejazdu.
Standardowa procedura	<ol style="list-style-type: none"> (1) Użytkownik przechodzi do widoku wyznaczania trasy. (2) Aplikacja pobiera dane o ulubionych miejscach z pliku lokalnego. (3) Użytkownik wybiera punkt startowy i końcowy trasy. (4) Użytkownik wybiera przystanki trasy. (opcjonalnie) (5) Użytkownik wybiera kryterium wyznaczania trasy. (6) Aplikacja wysyła zapytanie do serwera. (7) System wyznacza żadaną trasę. (8) Serwer zwraca wyznaczoną trasę do aplikacji. (9) Wyświetlona zostaje mapa z wyrysowaną trasą. (10) Informacje o trasie zostają zapisane w pliku lokalnym.
Alternatywne procedury	<ol style="list-style-type: none"> (2) Plik z ulubionymi miejscami jest niedostępny. (3) Wyświetlony zostaje chwilowo komunikat o błędzie. (4) Następuje kontynuacja procedury. (6) Aplikacja nie ma dostępu do internetu. (7) Wyświetlony zostaje widok z informacją o błędzie. (7) Jeden z niezbędnych komponentów jest niedostępny. (8) Serwer zwraca komunikat o błędzie do aplikacji. (9) Wyświetlony zostaje widok z informacją o błędzie. (10) Plik z historią tras jest niedostępny. (11) Wyświetlony zostaje chwilowo komunikat o błędzie.

Nazwa	Ustawienie stanu demonstracyjnego
Opis	W stanie demonstracyjnym system korzysta ze statycznych danych zapisanych w pamięci i jest niezależny od zewnętrznych komponentów. Daje to możliwość testowania i demonstracji aplikacji w kontrolowanych warunkach, nawet gdy system wypożyczania rowerów jest zamknięty.
Warunek wstępny	Administrator jest zalogowany.
Warunek końcowy	System zostaje wprowadzony w stan demonstracyjny.
Potencjalne wyjątki	1. Brak połączenia aplikacji z internetem. 2. Niedostępność jednego z niezbędnych komponentów.
Aktorzy	Administrator
Wyzwalacz	Administrator chce uniezależnić system od rzeczywistych danych na potrzeby demonstracji lub testów.
Standardowa procedura	(1) Administrator przechodzi do widoku zarządzania stanem systemu. (2) Administrator wybiera stan demonstracyjny. (3) Administrator potwierdza ustawienie stanu systemu. (4) Aplikacja wysyła zapytanie do serwera. (5) System przechodzi do stanu demonstracyjnego. (6) Serwer zwraca potwierdzenie ustawienia stanu. (7) Aplikacja potwierdza powodzenie operacji.
Alternatywne procedury	(4) Aplikacja nie ma dostępu do internetu. (5) Wyświetlony zostaje widok z informacją o błędzie. (5) Jeden z niezbędnych komponentów jest niedostępny. (6) Serwer zwraca komunikat o błędzie do aplikacji. (7) Wyświetlony zostaje widok z informacją o błędzie.

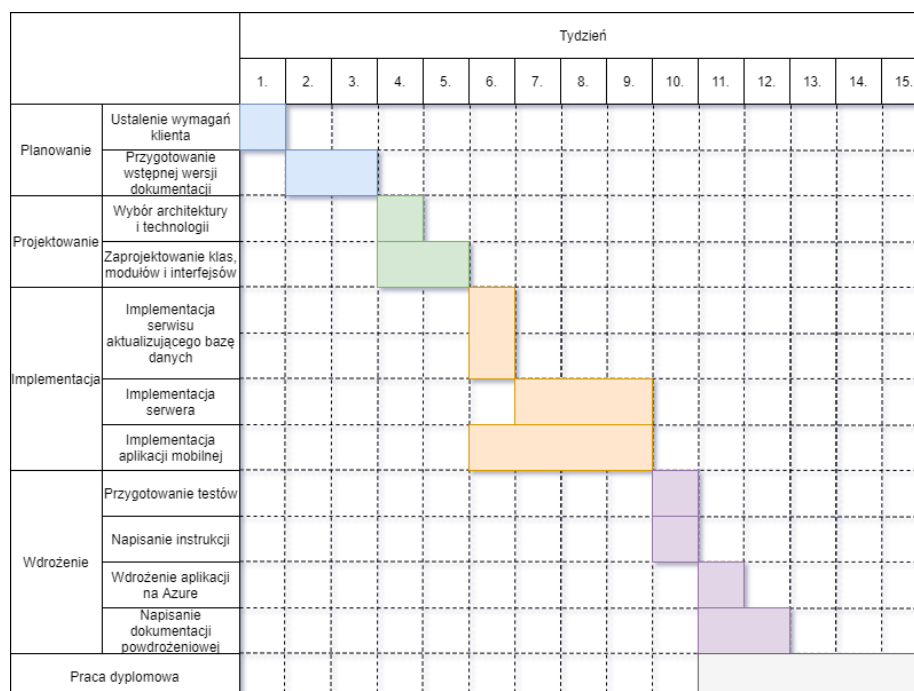
5 Wymagania niefunkcjonalne

Wymagania niefunkcjonalne przedstawia poniższa tabela.

Obszar wymagań	Opis wymagania
Użyteczność	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.
	Aplikacja mobilna jest kompatybilna z Androidem 9 oraz każdym nowszym.
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.

6 Planowanie

Na projekt zostanie przeznaczone 15 tygodni. Chcemy wykorzystać ten czas jak najefektywniej, dlatego zaplanowaliśmy prace na każdy tydzień. Rysunek 2 poniżej zawiera plan pracy.



Rysunek 2: 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.

7 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

8 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

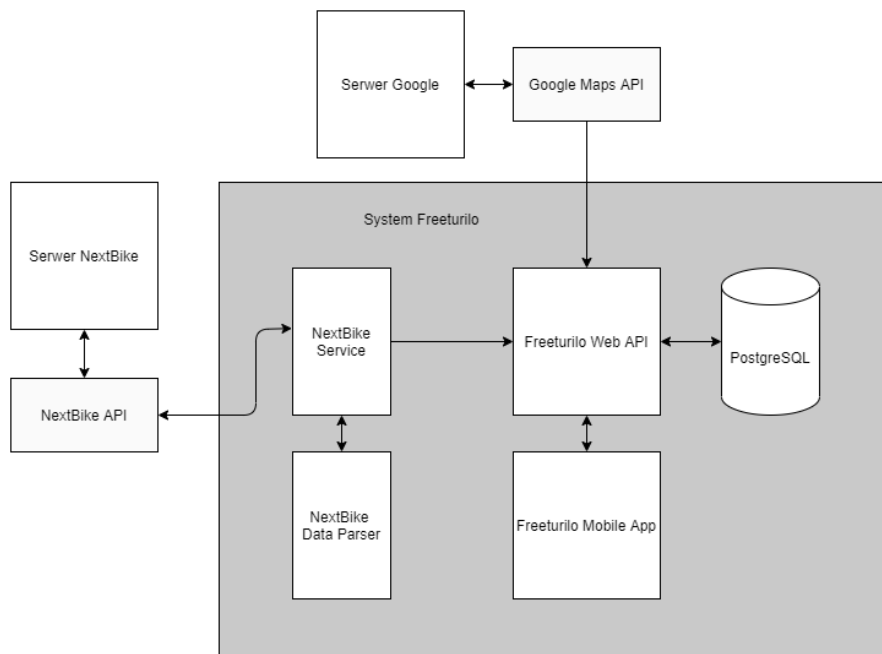
Realizacja projektu jest związana z zagrożeniami. Największym z nich jest zmiana publicznego interfejsu, który dostarcza informacji o lokalizacji rowerów. Spowodowałaby ona nieprawidłowe działanie aplikacji *Freeturilo* i zmusiłaby nas do natychmiastowej aktualizacji. Jest to ryzyko, przed którym trudno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

Inaczej jest w przypadku braku możliwości testowania aplikacji w okresie zimowym. Wśród funkcjonalności administratora znajdzie się wprowadzenie systemu w stan demonstracyjny, w którym korzysta on z danych testowych przygotowanych przez nas w okresie jesiennym. Aplikacja w takim stanie będzie mogła być bezproblemowo testowana, więc zabezpiecza nas to przed wspomnianym zagrożeniem.

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

9 Architektura

Rysunek 3 przedstawia planowany schemat architektury systemu *Freeturilo*.



Rysunek 3: Schemat architektury systemu

9.1 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.

9.2 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.

9.3 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.

Literatura

- [1] Dokumentacja systemu Android - <https://developer.android.com/>
- [2] Dokumentacja API Google Maps - <https://developers.google.com/maps/documentation>