



POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

Wydział Informatyki

Katedra Baz Danych

specjalizacja Bazy Danych

Katarzyna Wartanowicz

s22074

Trail Composer

Praca inżynierska

Paweł Lenkiewicz

miejsce, miesiąc, rok obrony

1 STRESZCZENIE I SŁOWA KLUCZOWE

2 SPIS TREŚCI

1	Streszczenie i słowa kluczowe	2
3	Skróty i symbole	5
4	Faza analizy	6
4.1	Wstęp	6
4.1.1	https://www.wikiloc.com/	6
4.1.2	https://velomapa.pl/	6
4.2	Wymagania funkcjonalne	7
4.2.1	Opis funkcjonalności systemu.....	7
4.2.2	Diagram przypadków użycia.....	8
4.2.3	Scenariusze przypadków użycia.....	14
4.3	Wymagania niefunkcjonalne	36
5	Faza projektowania	37
5.1	Opis wybranych technologii z uzasadnieniem	37
5.1.1	MS SQL.....	37
5.1.2	ASP.NET Core	37
5.1.3	Entity Framework.....	38
5.1.4	C#	39
5.1.5	React.....	40
5.1.6	React Router	41
5.1.7	Reactstrap	41
5.1.8	React Leaflet	42
5.1.9	Leaflet GPX plugin	43
5.1.10	JWT	44
5.1.11	Azure Active Directory B2C.....	45
5.1.12	MSAL React.....	46

5.2	Architektura systemu	47
5.2.1	Założenia do projektowania architektury systemu:	47
5.2.2	Opis architektury aplikacji w środowisku rozwojowym	47
5.2.3	Opis architektury aplikacji w środowisku produkcyjnym	49
5.2.4	Moduły aplikacji	50
5.3	Projekt interfejsu	51
5.4	Diagram związków encji z opisem	52
5.4.1	Tabela Trail	53
5.4.2	Tabela TCUser	54
5.4.3	Tabela PathType	54
5.4.4	Tabela PathLevel	54
5.4.5	Tabela Trail_Type	55
5.4.6	Tabela Segment	55
5.4.7	Tabela Trail_Segment	57
5.4.8	Tabela POI	58
5.4.9	Tabela Segment_POI	59
5.4.10	Tabela Segment_Type	59
5.4.11	Tabela POIType	60
5.4.12	Tabela POI_POIType	60
5.4.13	Tabela Country	61
5.4.14	Tabela Trail_Country	61
5.4.15	Tabela POIPhoto	62
5.5	Wymagania sprzętowe	63
5.5.1	Wymagania sprzętowe dla środowiska rozwojowego	63
	Wymagania sprzętowe dla środowiska produkcyjnego	64
6	Wyniki i dyskusja	68

7	Bibliografia	68
8	Załączniki.....	72

3 SKRÓTY I SYMBOLE

POI – od eng. Point of Interest, miejsca warte uwagi

GPX – od eng. GPS Exchange Format, zstandaryzowany schemat XML ułatwiający wymianę danych nawigacji satelitarnej

UD – od eng. Use Case Diagram, diagram przypadków użycia

UC – od eng. Use Case, przypadek użycia

EF – Entity Framework

JW. – JSON Web Token

4 FAZA ANALIZY

4.1 WSTĘP

Celem aplikacji Trails Composer jest umożliwienie użytkownikom współdzielenia doświadczeń z wycieczek turystycznych.

Użytkownicy mogą nagrywać przebieg wycieczki za pomocą urządzeń mobilnych i zapisywać w formacie GPX. Nasza platforma będzie pośredniczyć w wymianie tych plików między użytkownikami, którzy będą mogli również dodawać dodatkowe informacje na temat przebiegu wycieczki.

Istnieje kilka aplikacji realizujących podobne funkcje do Trails Composer, ale nierealizujące wszystkich funkcjonalności, jak wspólne tworzenie dłuższych tras na podstawie doświadczeń innych turystów.

4.1.1 <https://www.wikiloc.com/>

Aplikacja webowa skupiająca społeczność turystyczną z całego świata i z różnych dziedzin turystycznych. Pozwala na dodawanie i dokumentowanie przebytych tras za pomocą plików GPX, a także pobieranie plików GPX wybranych tras oraz udostępnianie innym użytkownikom dodanych przez siebie tras oraz punktów szczególnych. Aplikacja obsługuje również aspekty społecznościowe jak obserwowanie innych użytkowników lub oznaczenie, które trasy zostały przebyte wspólnie z innymi użytkownikami.

4.1.2 <https://velomapa.pl/>

Aplikacja zorientowana przede wszystkim na polskich rowerzystów. Także obsługuje wymianę plików GPX pomiędzy użytkownikami. Oprócz tego istnieje opcja dodawania wydarzeń związanych z kolarstwem.

4.2 WYMAGANIA FUNKCJONALNE

4.2.1 Opis funkcjonalności systemu

Trails Composer umożliwi stworzenie trasy skomponowanej z fragmentów (odcinków) udostępnionych przez użytkowników. Trasa będzie składała się z co najmniej jednego odcinka. Liczba odcinków przypisanych do trasy nie będzie mogła przekraczać 500.

Odcinek będzie tworzony poprzez wgranie pliku GPX, poziomu trudności, typu trasy oraz kraju, przez który przebiega oraz opcjonalnie dodanie opisu. Długość odcinka będzie wyliczana na podstawie pliku GPX. Opcjonalnie będzie można też dodać listę maksymalnie 100 POI występujących po drodze. Użytkownik będzie tworzył POI, zawierające współrzędne geograficzne, rodzaje POI oraz opcjonalnie krótki opis i zdjęcia. Będą dostępne trzy poziomy trudności, a typy tras będą dzielić się na piesze i rowerowe.

Poziom trasy, długość trasy, typy trasy oraz kraje, przez które przebiega trasa będą wyznaczone na podstawie odcinków, z których się składa.

Użytkownik będzie mógł założyć konto użytkownika. Do założenia konta będą niezbędne hasło, login oraz adres email.

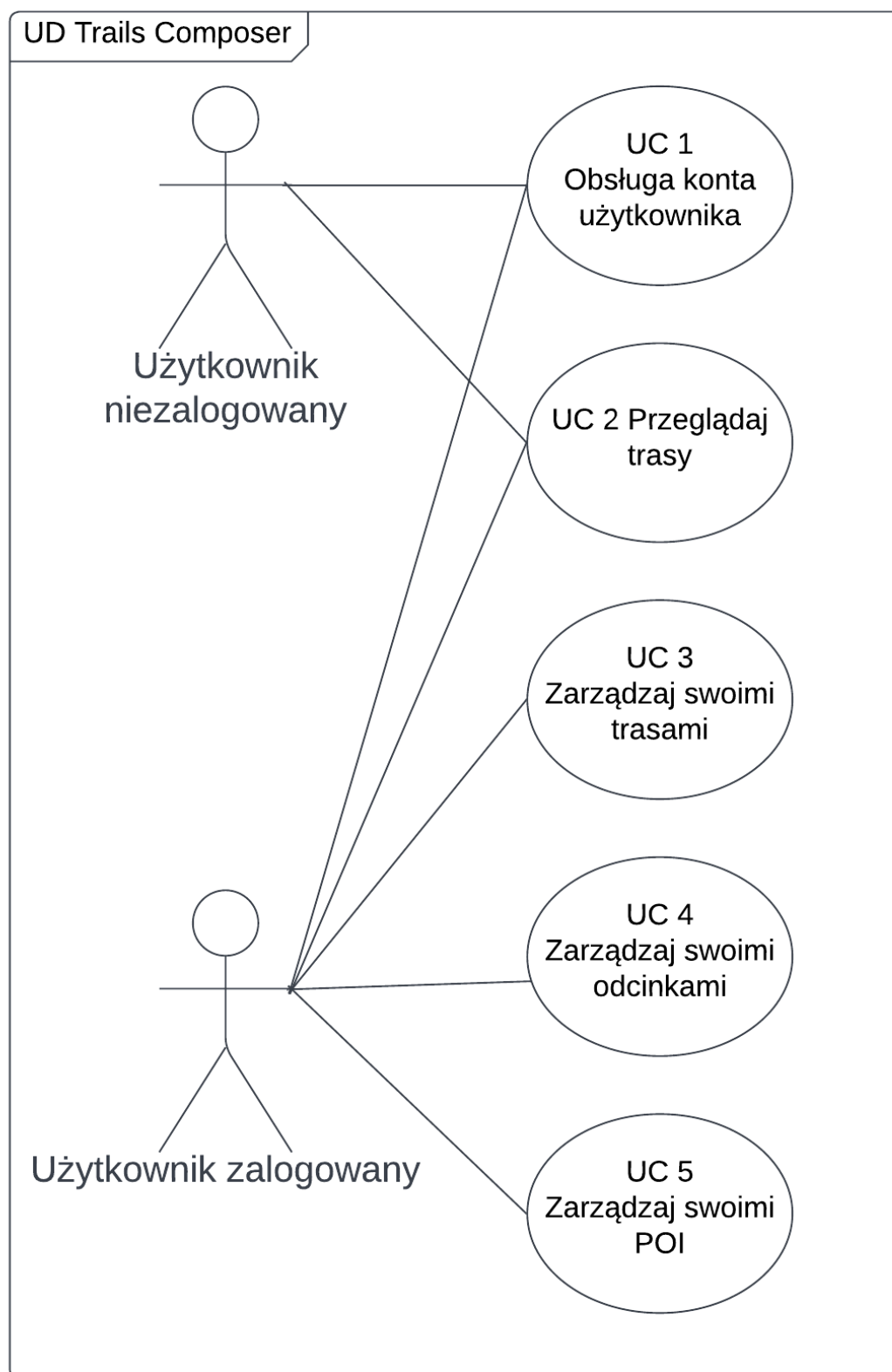
Aby móc tworzyć trasy, odcinki i POI użytkownik będzie musiał być zalogowany.

Zalogowany użytkownik będzie mógł wyświetlić listę dodanych przez siebie tras, odcinków i POI oraz aby pobrać plik GPX dowolnego wybranego odcinka, w tym również odcinków wprowadzonych przez innych użytkowników.

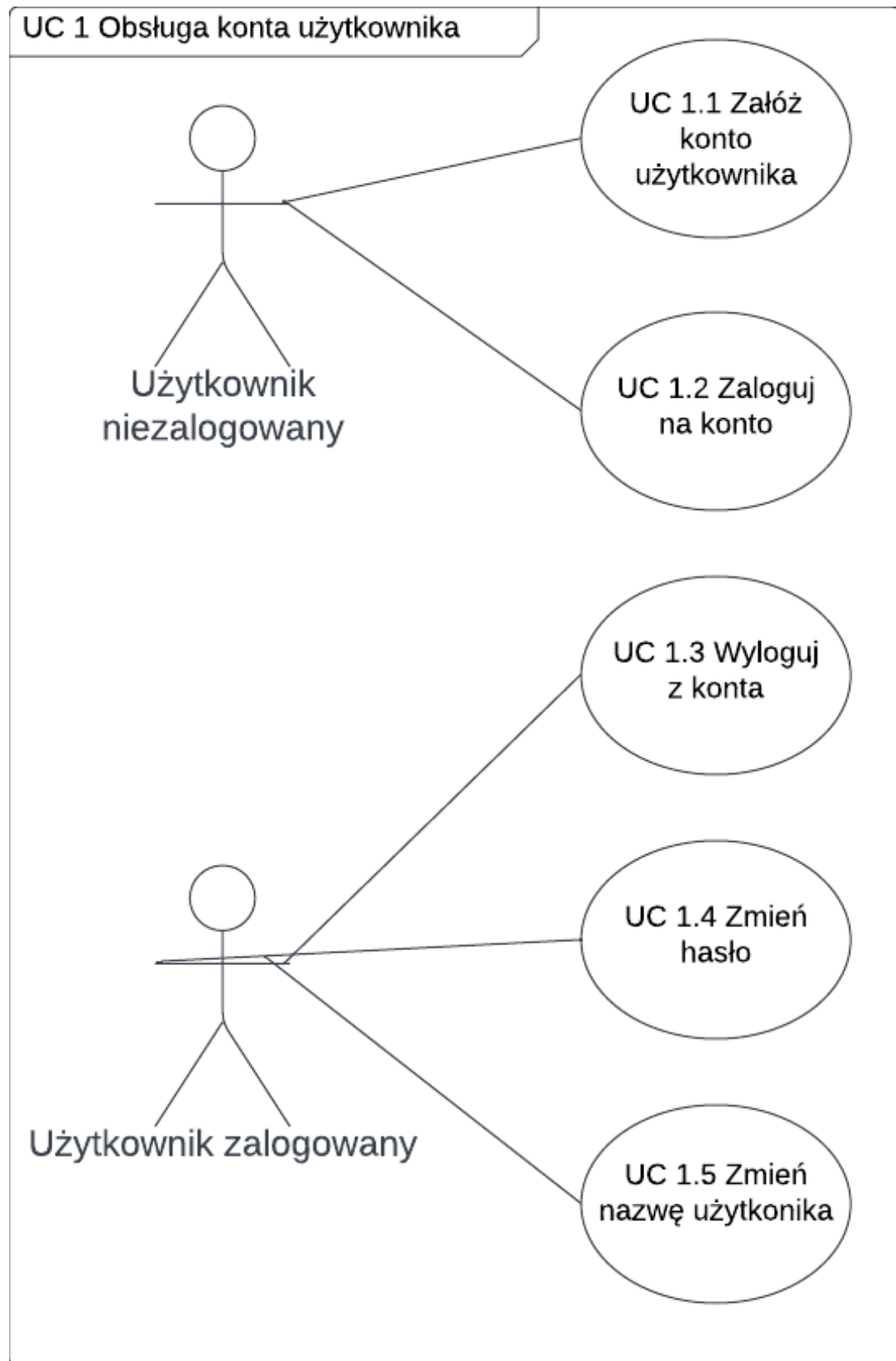
Użytkownik będzie mógł usunąć wszystkie elementy, których jest autorem. W przypadku odcinków i POI wykorzystywanych przez innych użytkowników wszystkie dane, które nie są niezbędne do zachowania integralności systemu są usuwane, a do listy typów usuniętego odcinka bądź POI będzie dodawany typ „usunięty”.

Użytkownik będzie mógł przeglądać trasy i odcinki innych użytkowników i filtrować je na podstawie poziomu trudności, typu trasy i nazwy kraju. Dla każdej trasy będzie można wyświetlić szczegółowe informacje o niej, odcinkach, z których się składa i POI z nimi powiązanymi oraz przebieg trasy na mapie. POI będą mogły być filtrowane na podstawie kraju, w którym się znajdują, oraz rodzajów.

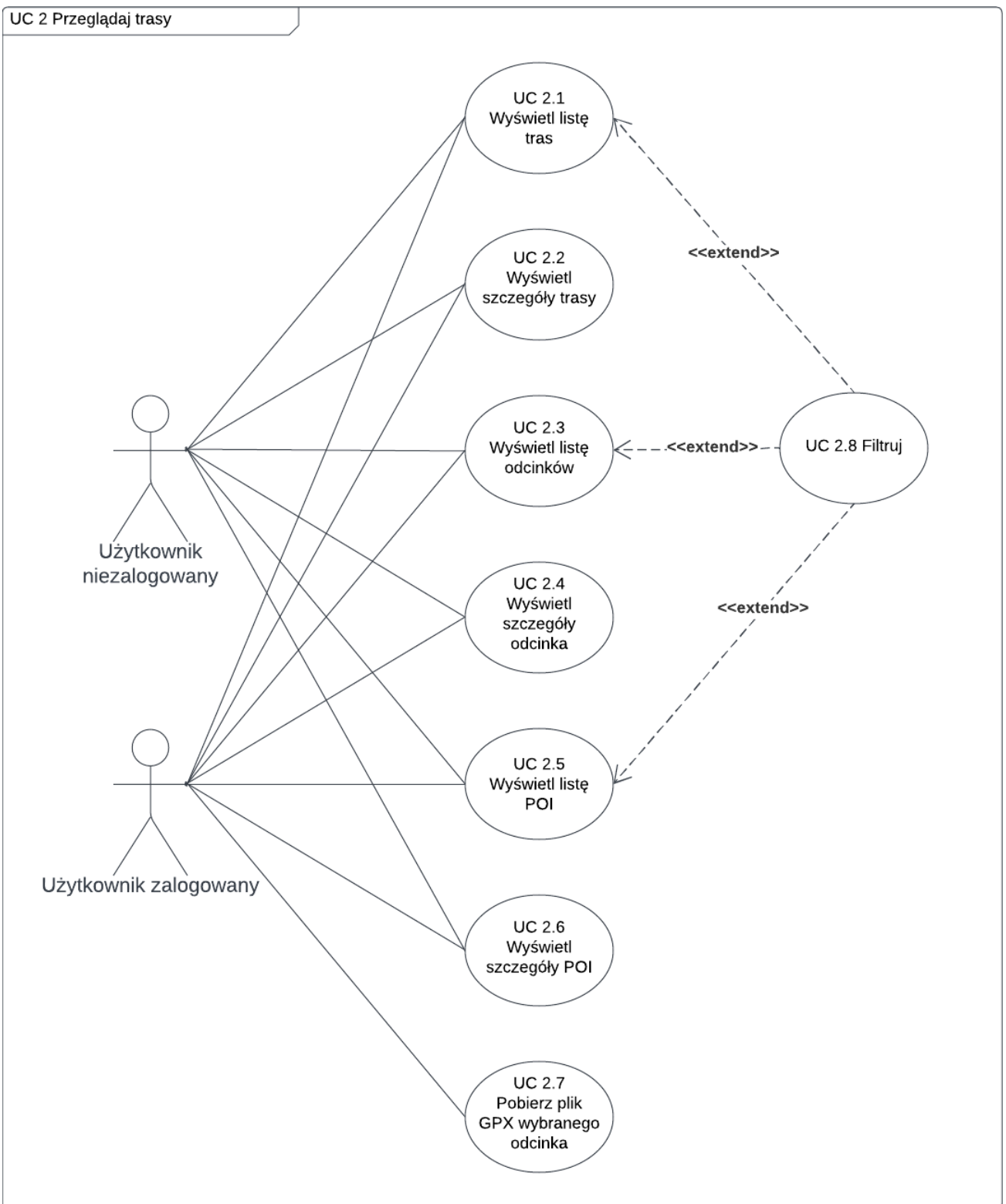
4.2.2 Diagram przypadków użycia



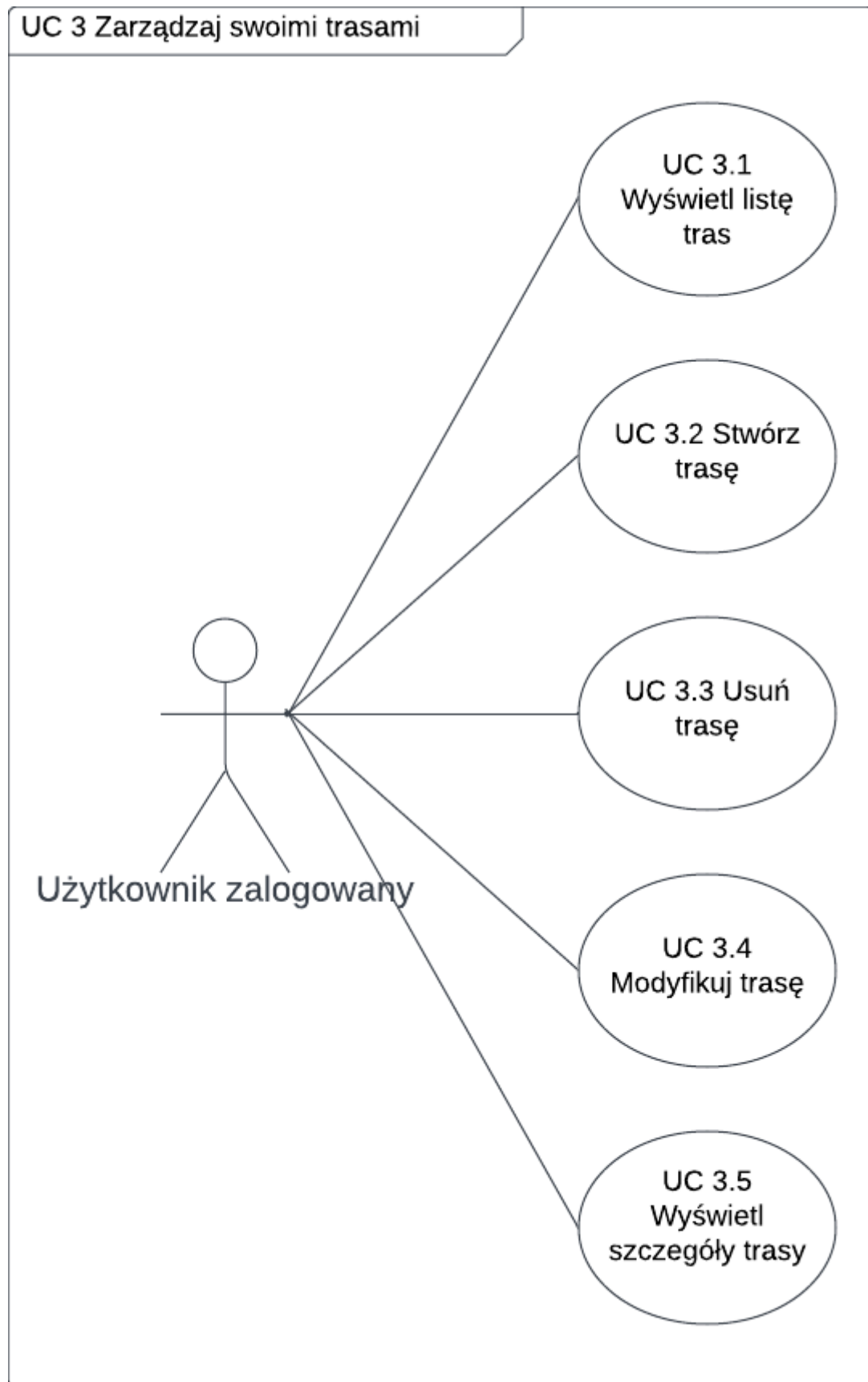
Rysunek 1. UD Trails Composer



Rysunek 2. UC 1 Obsługa konta użytkownika

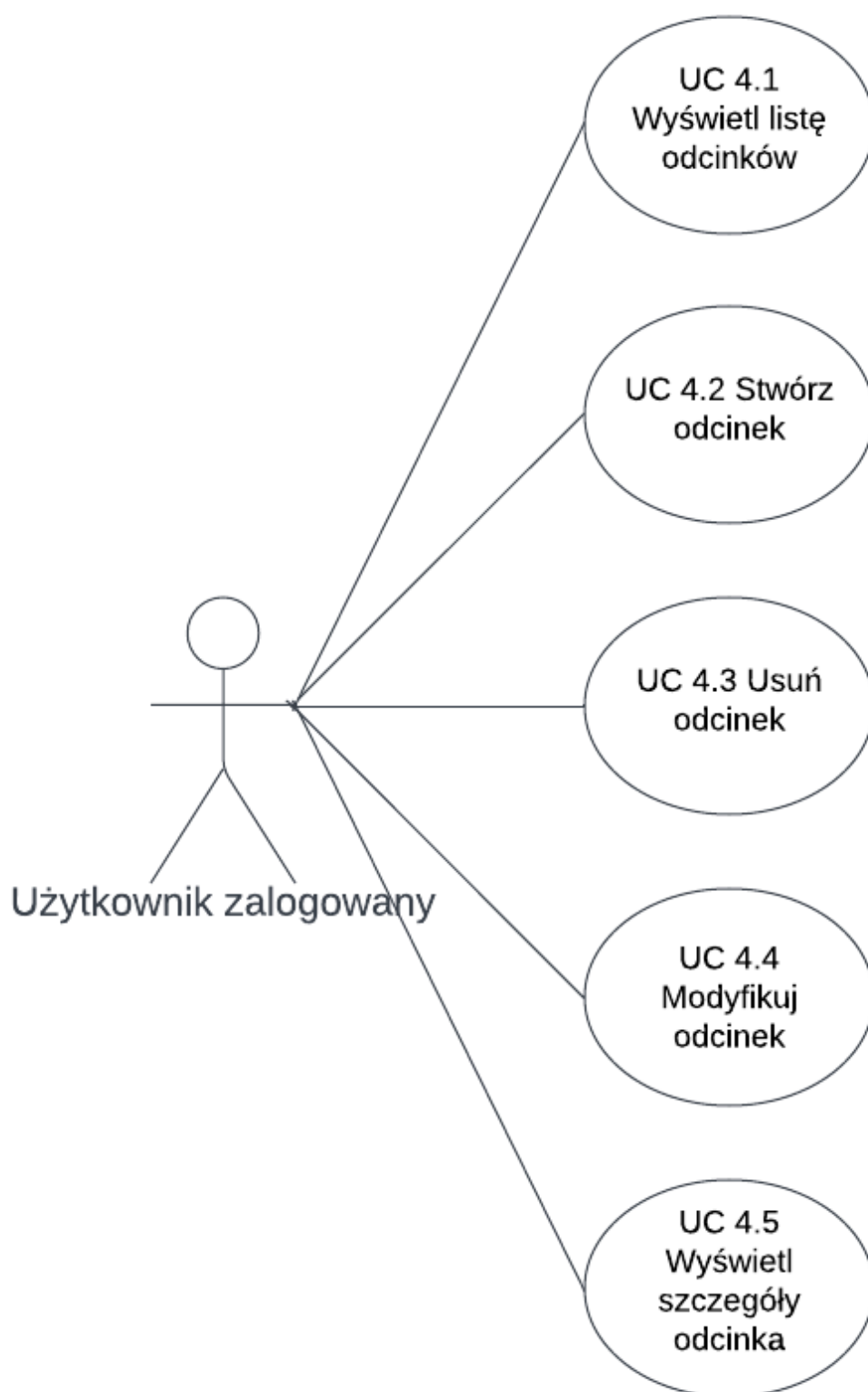


Rysunek 3. UC 2 Przeglądaj trasy

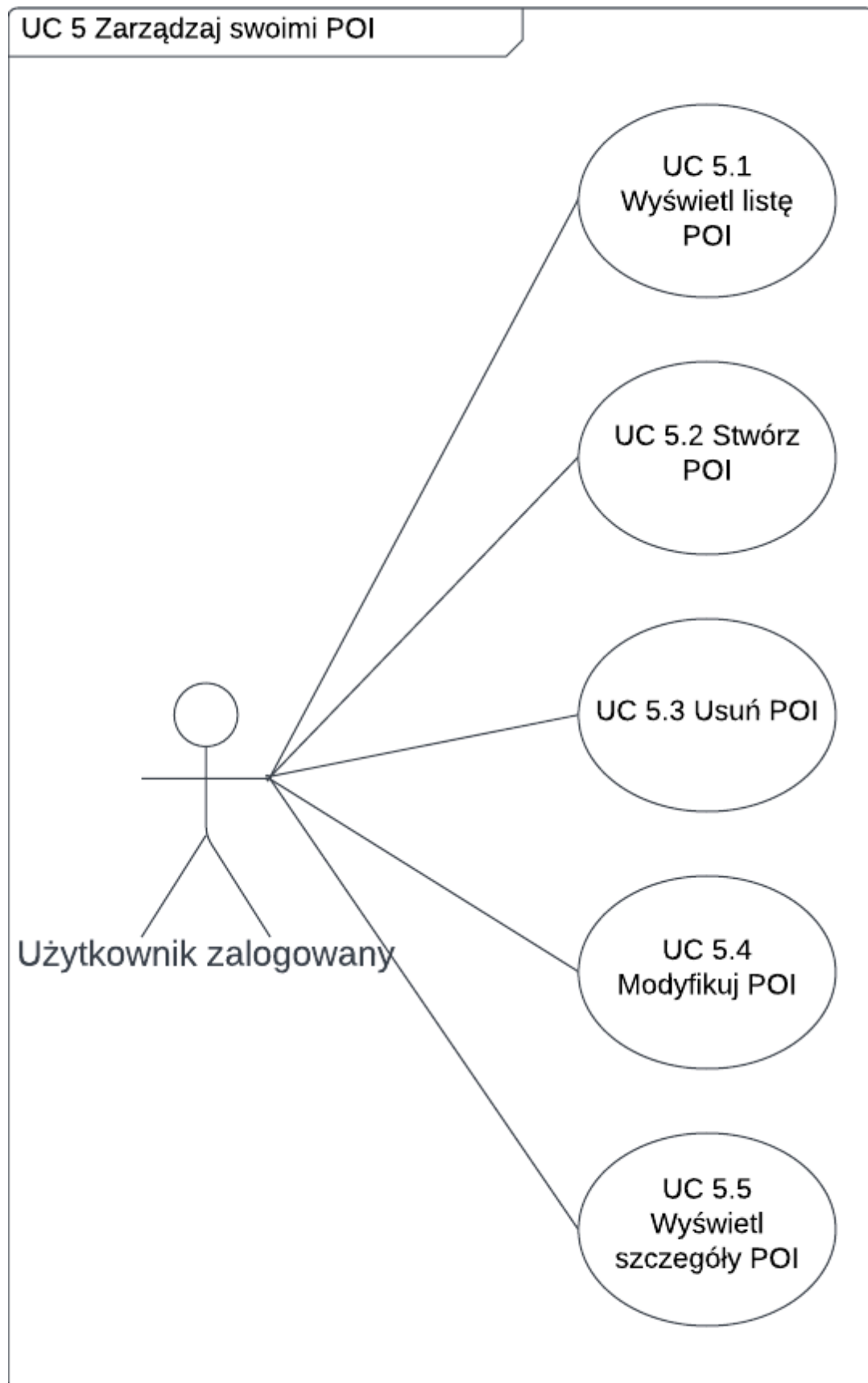


Rysunek 4. UC 3 Zarządzaj swoimi trasami

UC 4 Zarządzaj swoimi odcinkami



Rysunek 5. UC 4 Zarządzaj swoimi odcinkami



Rysunek 6. UC 5 Zarządzaj swoimi POI

4.2.3 Scenariusze przypadków użycia

4.2.3.1 UC 1 Obsługa konta użytkownika

Nazwa przypadku użycia	UC 1.1 Załóż konto użytkownika
Warunek początkowy	Użytkownik nie jest zalogowany na koncie.
Główny przepływ zdarzeń	<ol style="list-style-type: none">1. Aktor „użytkownik niezalogowany” uruchamia przypadek użycia.2. System wyświetla formularz zakładania konta.3. Aktor wypełnia formularz.4. System waliduje formularz.5. System informuje aktora o utworzeniu konta, przekierowuje i kończy przypadek użycia.
Alternatywne przepływy zdarzeń	4a. Wypełniony formularz nie jest poprawny: system informuje aktora o błędach formularza. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	System utworzył nowe konto użytkownika.

Tabela 1. Scenariusz UC 1.1 Załóż konto użytkownika

Nazwa przypadku użycia	UC 1.2 Zaloguj na konto
Warunek początkowy	Użytkownik nie jest zalogowany na koncie.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik niezalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz logowania. 3. Aktor wypełnia formularz. 4. System waliduje formularz. 5. System uwierzytelnia aktora, rozpoczyna sesję zalogowanego użytkownika i przekierowuje go na stronę główną. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	<p>4a. Wypełniony formularz nie jest poprawny: system informuje aktora o błędach formularza. Przypadek użycia przechodzi do punktu 3.</p> <p>5a. Uwierzytelnienie aktora nie jest możliwe: system informuje aktora o niemożliwości zalogowania. Przypadek użycia dobiega końca.</p>
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	Aktor jest uwierzytelniony i staje się „użytkownikiem zalogowanym”.

Tabela 2. Scenariusz UC 1.2 Zaloguj na konto

Nazwa przypadku użycia	UC 1.3 Wyloguj z konta
Warunek początkowy	Użytkownik jest zalogowany na koncie.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System kończy sesję zalogowanego użytkownika i przekierowuje aktora na stronę główną. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	brak
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Użytkownik nie jest zalogowany na koncie.

Tabela 3. Scenariusz UC 1.3 Wyloguj z konta

Nazwa przypadku użycia	UC 1.4 Zmień hasło
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz zmiany hasła. 3. Aktor uzupełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnej zmianie hasła i przekierowuje go na stronę z opcjami ustawień konta. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Wypełniony formularz nie jest poprawny: system informuje aktora o błędach formularza. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	Hasło użytkownika uległo zmianie.

Tabela 4. Scenariusz UC 1.4 Zmień hasło

Nazwa przypadku użycia	UC 1.5 Zmień nazwę użytkownika
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz zmiany nazwy użytkownika. 3. Aktor uzupełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnej zmianie nazwy użytkownika i przekierowuje go na stronę z opcjami ustawień konta. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Wypełniony formularz nie jest poprawny: system informuje aktora o błędach formularza. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	Nazwa użytkownika uległa zmianie.

Tabela 5. Scenariusz UC 1.5 Zmień nazwę użytkownika

4.2.3.2 UC 2 Przeglądaj trasy

Nazwa przypadku użycia	UC 2.1 Wyświetl listę tras
Warunek początkowy	Brak.
Główny przepływ zdarzeń	<ol style="list-style-type: none">1. Aktor „użytkownik niezalogowany” lub aktor „użytkownik zalogowany” uruchamia przypadek użycia.2. System wyświetla kilka pierwszych tras z listy posortowanej alfabetycznie po nazwach tras. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie tras nie jest możliwe z powodu błędu. System wyświetla odpowiedni komunikat o błędzie. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi listę tras.

Tabela 6. Scenariusz UC 2.1 Wyświetl listę tras

Nazwa przypadku użycia	UC 2.2 Wyświetl szczegóły trasy
Warunek początkowy	Aktor widzi listę tras, opcjonalnie przefiltrowaną.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik niezalogowany” lub aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla nazwę, poziom, typy, całkowitą długość i opis trasy oraz listę krajów, przez które biegnie trasa i prezentuje trasę na mapie. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie informacji o trasie nie jest możliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi metadane trasy oraz mapę z wyrysowanym przebiegiem trasy.

Tabela 7. Scenariusz UC 2.2 Wyświetl szczegóły trasy

Nazwa przypadku użycia	UC 2.3 Wyświetl listę odcinków
Warunek początkowy	Aktor widzi szczegóły trasy, której listę odcinków chce wyświetlić.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik niezalogowany” lub aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla kilka pierwszych odcinków z listy. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie odcinków nie jest możliwe z powodu błędu. System wyświetla odpowiedni komunikat o błędzie. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi listę odcinków.

Tabela 8. Scenariusz UC 2.3 Wyświetl listę odcinków

Nazwa przypadku użycia	UC 2.4 Wyświetl szczegóły odcinka
Warunek początkowy	Aktor widzi listę odcinków.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik niezalogowany” lub aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla nazwę, poziom, typ, długość i opis odcinka oraz jego reprezentację na mapie. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie informacji o odcinku nie jest możliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi metadane odcinka oraz mapę z wyrysowanym przebiegiem odcinka.

Tabela 9. Scenariusz UC 2.4 Wyświetl szczegóły odcinka

Nazwa przypadku użycia	UC 2.5 Wyświetl listę POI
Warunek początkowy	Aktor widzi szczegóły trasy lub odcinka.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik niezalogowany” lub aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla kilka pierwszych POI z listy. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	<p>2a. Wyświetlenie POI nie jest możliwe z powodu błędu. System wyświetla odpowiedni komunikat o błędzie. Przypadek użycia dobiega końca.</p>
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi listę POI.

Tabela 10. Scenariusz UC 2.5 Wyświetl listę POI

Nazwa przypadku użycia	UC 2.6 Wyświetl szczegóły POI
Warunek początkowy	Aktor widzi listę POI.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik niezalogowany” lub aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla nazwę, typ, opis, długość geograficzną, szerokość geograficzną i kraj, w którym leży POI oraz jedno ze zdjęć POI. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie informacji o POI nie jest możliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi metadane POI oraz zdjęcia z nim powiązane.

Tabela 11. Scenariusz UC 2.6 Wyświetl szczegóły POI

Nazwa przypadku użycia	UC 2.7 Pobierz plik GPX wybranego odcinka
Warunek początkowy	Aktor widzi szczegóły odcinka, który chce pobrać.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia poprzez wybranie odcinka, którego plik GPX chce pobrać. 2. System przygotowuje odpowiedni plik GPX do pobrania. 3. Aktor pobiera plik GPX. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Plik GPX wybranego odcinka nie jest dostępny. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor jest w posiadaniu pliku GPX wybranego odcinka.

Tabela 12. Scenariusz UC 2.7 Pobierz plik GPX wybranego odcinka

Nazwa przypadku użycia	UC 2.8 Filtruj
Warunek początkowy	Aktor widzi listę wszystkich tras, odcinków lub POI.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik niezalogowany” lub aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz z opcjami filtrowania. 3. Aktor wybiera opcje filtrowania. 4. System wyświetla kilka pierwszych pozycji z przefiltrowanej listy. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Wyświetlenie listy nie jest możliwe. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi przefiltrowaną listę.

Tabela 13. Scenariusz UC 2.8 Filtruj

4.2.3.3 UC 3 Zarządzaj swoimi trasami

Nazwa przypadku użycia	UC 3.1 Wyświetl listę własnych tras
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none">1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia.2. System wyświetla kilka pierwszych pozycji z listy tras, których właścicielem jest aktor. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie tras jest niemożliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi listę tras, których jest właścicielem.

Tabela 14. Scenariusz UC 3.1 Wyświetl listę własnych tras

Nazwa przypadku użycia	UC 3.2 Stwórz trasę
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz dodawania trasy. 3. Aktor wypełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnym utworzeniu trasy. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Formularz zawiera błędy lub utworzenie trasy jest w danym momencie niemożliwe. System wyświetla komunikat o odpowiednim błędzie. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	W systemie dostępna jest nowa trasa.

Tabela 15. Scenariusz UC 3.2 Stwórz trasę

Nazwa przypadku użycia	UC 3.3 Usuń trasę
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System usuwa trasę z systemu.
Alternatywne przepływy zdarzeń	Brak.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Usunięta trasa nie jest dostępna do wglądu.

Tabela 16. Scenariusz UC 3.3 Usuń trasę

Nazwa przypadku użycia	UC 3.4 Modyfikuj trasę
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz edycji trasy. 3. Aktor wypełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnej modyfikacji trasy. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Formularz zawiera błędy lub modyfikacja trasy jest w danym momencie niemożliwe. System wyświetla komunikat o odpowiednim błędzie. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	Dane wybranej trasy uległy zmianie.

Tabela 17. Scenariusz UC 3.4 Modyfikuj trasę

Nazwa przypadku użycia	UC 3.5 Wyświetl szczegóły trasy
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla metadane trasy oraz jej przebieg na mapie. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie szczegółów trasy jest niemożliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi szczegóły trasy.

Tabela 18. Scenariusz UC 3.5 Wyświetl szczegóły trasy

4.2.3.4 UC 4 Zarządzaj swoimi odcinkami

Nazwa przypadku użycia	UC 4.1 Wyświetl listę własnych odcinków
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla kilka pierwszych pozycji z listy odcinków, których właścicielem jest aktor. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie odcinków jest niemożliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi listę odcinków, których jest właścicielem.

Tabela 19. Scenariusz UC 4.1 Wyświetl listę własnych odcinków

Nazwa przypadku użycia	UC 4.2 Stwórz odcinek
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz dodawania odcinka. 3. Aktor wypełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnym utworzeniu odcinka. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Formularz zawiera błędy lub utworzenie odcinka jest w danym momencie niemożliwe. System wyświetla komunikat o odpowiednim błędzie. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	W systemie dostępna jest nowy odcinek.

Tabela 20. Scenariusz UC 4.2 Stwórz odcinek

Nazwa przypadku użycia	UC 4.3 Usuń odcinek
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System usuwa odcinek z systemu.
Alternatywne przepływy zdarzeń	Brak.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Usunięty odcinek nie jest dostępna do wglądu.

Tabela 21. Scenariusz 4.3 Usuń odcinek

Nazwa przypadku użycia	UC 4.4 Modyfikuj odcinek
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz edycji odcinka. 3. Aktor wypełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnej modyfikacji odcinka. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Formularz zawiera błędy lub modyfikacja odcinka jest w danym momencie niemożliwe. System wyświetla komunikat o odpowiednim błędzie. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	Dane wybranego odcinka uległy zmianie.

Tabela 22. Scenariusz UC 4.4 Modyfikuj odcinek

Nazwa przypadku użycia	UC 4.5 Wyświetl szczegóły odcinka
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla metadane odcinka oraz jej przebieg na mapie. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie szczegółów odcinka jest niemożliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi szczegóły odcinka.

Tabela 23. Scenariusz UC 4.5 Wyświetl szczegóły odcinek

4.2.3.5 UC 5 Zarządzaj swoimi POI

Nazwa przypadku użycia	UC 5.1 Wyświetl listę własnych POI
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla kilka pierwszych pozycji z listy POI, których właścicielem jest aktor. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie POI jest niemożliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi listę POI, których jest właścicielem.

Tabela 24. Scenariusz 5.1 Wyświetl listę własnych POI

Nazwa przypadku użycia	UC 5.2 Stwórz POI
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz dodawania POI. 3. Aktor wypełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnym utworzeniu POI. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Formularz zawiera błędy lub utworzenie POI jest w danym momencie niemożliwe. System wyświetla komunikat o odpowiednim błędzie. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	W systemie dostępna jest nowe POI.

Tabela 25. Scenariusz 5.2 Stwórz POI

Nazwa przypadku użycia	UC 5.3 Usuń POI
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System usuwa POI z systemu.
Alternatywne przepływy zdarzeń	Brak.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Usunięte POI nie jest dostępna do wglądu.

Tabela 26. Scenariusz UC 5.3 Usuń POI

Nazwa przypadku użycia	UC 5.4 Modyfikuj POI
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla formularz edycji POI. 3. Aktor wypełnia formularz. 4. System waliduje formularz. 5. System informuje aktora o pomyślnej modyfikacji POI. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	4a. Formularz zawiera błędy lub modyfikacja POI jest w danym momencie niemożliwe. System wyświetla komunikat o odpowiednim błędzie. Przypadek użycia przechodzi do punktu 3.
Zakończenie	Zgodnie ze scenariuszem lub na żądanie aktora.
Warunek końcowy	Dane wybranego POI uległy zmianie.

Tabela 27. Scenariusz UC 5.4 Modyfikuj POI

Nazwa przypadku użycia	UC 5.5 Wyświetl szczegóły POI
Warunek początkowy	Użytkownik jest zalogowany.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor „użytkownik zalogowany” uruchamia przypadek użycia. 2. System wyświetla metadane POI. Przypadek użycia dobiega końca.
Alternatywne przepływy zdarzeń	2a. Wyświetlenie szczegółów POI jest niemożliwe z powodu błędu. System wyświetla odpowiedni komunikat. Przypadek użycia dobiega końca.
Zakończenie	Zgodnie ze scenariuszem.
Warunek końcowy	Aktor widzi szczegóły POI.

Tabela 28. Scenariusz UC 5.5 Wyświetl szczegóły POI

4.3 WYMAGANIA NIEFUNKCJONALNE

Trails Composer będzie aplikacją webową działającą na współczesnych wersjach przeglądarek: Google Chrome, Microsoft Edge i Opera.

Wszystkie trasy, odcinki, POI oraz zdjęcia z nimi powiązane będą udostępniane tylko i wyłącznie na użytek niekomercyjny.

Aplikacja będzie spełniała wymagania Unii Europejskiej w zakresie ochrony danych osobowych.

Aplikacja będzie korzystała z uwierzytelniania opartego na plikach cookie.

Czas odpowiedzi systemu w 90% przypadków będzie poniżej pięciu sekund (nie dotyczy stron zawierających mapę) przy obciążeniu jednego żądania na sekundę. Maksymalna liczba żądań na sekundę to trzy.

System będzie obsługiwał pliki w formacie GPX o rozmiarze nie większym niż 500 punktów, aby zachować kompatybilność z większością urządzeń GPS.

System będzie przyjmuje zdjęcia w formatach jpg o rozmiarze nie większym niż 10 MB.

5 FAZA PROJEKTOWANIA

5.1 OPIS WYBRANYCH TECHNOLOGII Z UZASADNIENIEM

5.1.1 MS SQL

Microsoft SQL Server, nazywany również MS SQL, to system zarządzania bazą danych relacyjnych (RDBMS) rozwijany przez firmę Microsoft. Jest to kompleksowa i skalowalna platforma bazodanowa powszechnie używana do zarządzania i przechowywania danych w różnych aplikacjach, od małych rozwiązań biznesowych do dużych systemów przedsiębiorstw.

NIEZAWODNOŚĆ I STABILNOŚĆ: MS SQL Server jest znany z niezawodności i stabilności. Posiada dojrzałą architekturę, rozbudowane funkcje i od lat szeroko stosowany jest w środowiskach korporacyjnych.

SKALOWALNOŚĆ: MS SQL Server pozostanie dobrym rozwiązaniem także wtedy, gdy aplikacja się rozrośnie

INTEGRACJA Z SYSTEMAMI MICROSOFT: MS SQL Server doskonale integruje się z innymi technologiami Microsoft, takimi jak usługi chmurowe Azure, framework .NET i Visual Studio co przyspieszy procesy rozwoju i wdrażania.

NARZĘDZIA ZARZĄDZANIA: SQL Server Management Studio dostarcza rozbudowany zestaw narzędzi do administracji bazą danych, co ułatwia zarządzanie nią i jej utrzymanie.

DOKUMENTACJA I WSPARCIE SPOŁECZNOŚCI: dzięki dużej społeczności użytkowników i wsparciu ze strony Microsoft, istnieje wiele zasobów, dokumentacji i forów społecznościowych do rozwiązywania problemów.

5.1.2 ASP.NET Core

ASP.NET Core to otwarto źródłowy, wieloplatformowy framework webowy opracowany przez Microsoft do tworzenia nowoczesnych, opartych na chmurze i skalowalnych aplikacji internetowych. Jest to modularny i wydajny framework, będący znaczącą ewolucją w porównaniu do wcześniejszego frameworka ASP.NET.

WIELOPLATFARMOWOŚĆ: ASP.NET Core działa na systemach Windows, macOS i Linux. Zapewni to elastyczność w kwestii wdrożeń i opcji hostingu.

MODUŁOWOŚĆ I MIDDLEWARE: ASP.NET Core promuje modułową i lekką architekturę. Koncepcja middleware umożliwia łatwe dodawanie, usuwanie lub konfigurowanie komponentów, co zwiększa elastyczność i łatwość utrzymania.

WBUDOWANE WSTRZYKIWANIE ZALEŻNOŚCI (DEPENDENCY INJECTION):

ASP.NET Core posiada wbudowane wsparcie dla wstrzykiwania zależności, co sprzyja bardziej zorganizowanemu i testowalnemu kodowi oraz zapewnia lepszą strukturę kodu i zarządzanie zależnościami.

INTEGRACJA Z NOWOCZESNYMI STANDARDAMI INTERNETOWYMI (WEB STANDARDS): ASP.NET Core doskonale integruje się z nowoczesnymi standardami i technologiami webowymi. Obsługuje WebSockets, interfejsy API JSON i inne współczesne funkcje, zapewniając, że aplikacja pozostanie zgodna z trendami branżowymi.

FUNKCJE BEZPIECZEŃSTWA: ASP.NET Core zawiera wbudowane funkcje zabezpieczeń, które pomagają chronić aplikacje. Zapewnia mechanizmy ochrony danych, uwierzytelniania i autoryzacji, co ułatwia budowanie bezpiecznych usług back-endowych.

NARZĘDZIA I ŚRODOWISKO PROGRAMISTYCZNE: ASP.NET Core jest dostarczany z bogatym zestawem narzędzi programistycznych, w tym obsługą Visual Studio i Visual Studio Code. To zapewnia płynne doświadczenie programistyczne i solidne możliwości debugowania.

WSPARCIE SPOŁECZNOŚCI: ASP.NET Core cieszy się silnym wsparciem ze strony społeczności. Oznacza to dostęp do obszernej dokumentacji, samouczków i forum społecznościowego, co ułatwia znalezienie rozwiązań dla ewentualnych problemów podczas tworzenia aplikacji.

5.1.3 Entity Framework

Entity Framework (EF) to framework do mapowania obiektowo-relacyjnego (ORM) opracowany przez Microsoft. Zapewnia zestaw narzędzi i bibliotek dla programistów .NET, umożliwiając pracę z bazami danych przy użyciu zasad obiektowo-zorientowanego programowania. Głównym celem Entity Framework jest uproszczenie procesu interakcji z bazą danych w aplikacjach .NET.

MAPOWANIE OBIEKTOWO-RELACYJNE (ORM): Entity Framework uprości interakcję między aplikacją a bazą danych.

PRODUKTYWNOŚĆ: EF redukuje ilość powtarzalnego kodu potrzebnego do interakcji z bazą danych, co pozwala skupić się na logice aplikacji.

INTEGRACJA Z LINQ: EF doskonale integruje się z Language Integrated Query (LINQ), co pozwala programistom pisać bezpieczne typy zapytań, używając języka programowania zamiast surowego SQL.

5.1.4 C#

C# to nowoczesny, wieloparadygmatowy język programowania opracowany przez firmę Microsoft w ramach platformy .NET. Po raz pierwszy został wprowadzony na początku lat 2000 i od tego czasu stał się jednym z głównych języków do budowania różnych rodzajów aplikacji, począwszy od oprogramowania desktopowego, poprzez aplikacje internetowe, aż po usługi oparte na chmurze.

WIELOPLATFORMOWOŚĆ I UNIWERSALNOŚĆ : C# jest wszechstronnym językiem programowania, który pozwala budować różnorodne aplikacje. C# jest wieloplatformowy, co oznacza, że może działać bezproblemowo na systemach Windows, macOS i Linux.

INTEGRACJA Z ASP.NET: C# bardzo dobrze się integruje z ASP.NET, stabilnym frameworkiem do tworzenia aplikacji internetowych. Używanie C# z ASP.NET zapewnia spójne i efektywne doświadczenie programistyczne.

SILNE TYPOWANIE I BEZPIECZEŃSTWO: silne typowanie w C# zwiększa bezpieczeństwo kodu i redukuje ilość błędów podczas działania programu. Stosowanie języka C# sprzyja stosowaniu najlepszych praktyk programistycznych i pomaga wykrywać potencjalne problemy podczas rozwoju oprogramowania.

BOGATA BIBLIOTEKA STANDARDOWA: C# posiada obszerną bibliotekę standardową, która dostarcza szeroki zakres narzędzi, bibliotek i frameworków do różnych potrzeb programistycznych. Zapewnia to wygodę korzystania z dobrze utrzymanego i spójnego rozwiązania, co zmniejsza potrzebę zależności od zewnętrznych bibliotek.

ZASADY PROGRAMOWANIA OBIEKTOWEGO (OOP): C# wspiera zasady programowania obiektowego, co czyni go doskonałym wyborem do tworzenia modularnego i łatwego do utrzymania kodu. Zasady OOP przyczyniają się do organizacji kodu i jego dogodnego ponownego użycia.

PRODUKTYWNOŚĆ PROGRAMISTY: w C# programuje się wykorzystaniem takich IDE jak Visual Studio, które zwiększają produktywność programisty i są wyposażone w

IntelliSense, wsparcie refaktoryzacji kodu i możliwości debugowania. Wiele funkcji języka pomaga pisać czytelny, zwięzły i ekspresyjny kod.

WSPARCIE MICROSOFTU: Microsoft zapewnia silne wsparcie, regularne aktualizacje i dużą liczbę dostępnych zasobów dla programistów.

5.1.5 React

React to biblioteka JavaScript do budowania interfejsów użytkownika opracowana i utrzymywana przez Facebooka. React zdobył ogromną popularność w społeczności deweloperów ze względu na swoją prostotę, wydajność i deklaratywne podejście do tworzenia komponentów interfejsu użytkownika.

SKŁADNIA DEKLARATYWNA: React używa składni deklaratywnej, co ułatwia zrozumienie i pisanie kodu. To prowadzi do bardziej czytelnego i łatwiejszego w utrzymaniu kodu, zmniejszając ryzyko popełniania błędów.

ARCHITEKTURA OPARTA NA KOMPONENTACH: Architektura oparta na komponentach promuje ponowne wykorzystanie i modularność. To pozwala na efektywny rozwój i łatwiejsze utrzymanie, ponieważ można zarządzać każdą częścią interfejsu użytkownika niezależnie.

WIRTUALNY DOM: Wirtualny DOM w React zapewnia optymalną wydajność poprzez aktualizowanie tylko niezbędnych komponentów, co przekłada się na szybsze renderowanie i płynniejsze doświadczenie użytkownika.

DUŻA I AKTYWNA SPOŁECZNOŚĆ: React cieszy się dużą i aktywną społecznością, co zapewnia szeroki dostęp do zasobów, bibliotek i wsparcia. To ułatwia rozwiązywanie problemów i pozwala być na bieżąco z najnowszymi trendami rozwoju oprogramowania.

WSPARCIE OD FACEBOOKA: Jako projekt rozwijany i utrzymywany przez Facebooka, React ciągle się rozwija i ma większe szanse na pozostanie aktualnym oraz gwarantuje częste aktualizacje w których są usuwane najbardziej dokuczliwe błędy.

5.1.6 React Router

Sam React nie ma wbudowanej obsługi routingu. Istnieją jednak popularne biblioteki innych firm implementujące tę funkcjonalność, które można użyć w aplikacjach React. Jedną z takich bibliotek jest React Router, która zapewnia kompleksowe rozwiązanie routingu. Umożliwia definiowanie tras, zarządzanie nawigacją i obsługę zmian adresów URL w sposób przyjazny dla React.

DEKLARATYWNA NAWIGACJA: React Router dostarcza deklaracyjny sposób definiowania struktury nawigacyjnej twojej aplikacji. Dzięki React Router możesz wyrażać trasy w składni opartej na JSX, co ułatwia zrozumienie i utrzymanie kodu.

STRUKTURA OPARTA NA KOMPONENTACH: React Router promuje podejście oparte na komponentach do nawigacji. Każda trasa odpowiada komponentowi React, umożliwiając zgrupowanie logiki i interfejsu użytkownika dla każdej trasy w modularny i wielokrotnego użycia sposób.

ZAGNIEŹDŻONA NAWIGACJA: React Router obsługuje zagnieżdżone trasy, umożliwiając tworzenie złożonych struktur interfejsu użytkownika z wieloma poziomami nawigacji. To ułatwia zarządzanie stanem i zachowaniem związanym z różnymi sekcjami aplikacji.

DYNAMICZNE TRASY: React Router umożliwia dynamiczne dopasowywanie tras. Możesz przekazywać parametry w adresie URL i uzyskiwać do nich dostęp wewnątrz swoich komponentów, co ułatwia tworzenie dynamicznych i opartych na danych widoków.

INTEGRACJA Z HISTORIĄ PRZEGLĄDARKI: React Router ładnie integruje się z interfejsem API historii przeglądarki. Oznacza to, że nawigacja aktualizuje adres URL bez konieczności pełnego ponownego ładowania strony, co zapewnia płynne doświadczenie użytkownika w SPA.

5.1.7 Reactstrap

Reactstrap to biblioteka, która udostępnia komponenty Bootstrap jako komponenty React, umożliwiając programistom korzystanie z stylizacji i komponentów Bootstrapa bezproblemowo w aplikacjach React. Bootstrap to popularny framework front-end, który ułatwia projektowanie i stylizację responsywnych stron internetowych.

INTEGRACJA Z BOOTSTRAPEM: Reactstrap to opakowanie (Wrapper) napisany w React dla popularnego frameworka CSS, Bootstrapa. Reactstrap, wykorzystuje moc i popularność Bootstrapa, zapewniając spójny i responsywny projekt aplikacji.

PONOWNE WYKORZYSTANIE KOMPONENTÓW: Reactstrap dostarcza zestaw gotowych do użycia, wysoko konfigurowalnych komponentów zgodnych z zasadami projektowymi Bootstrapa. To sprzyja ponownemu wykorzystywaniu komponentów, oszczędzając czas i wysiłek podczas programowania.

RESPANSYWNE PROJEKTOWANIE STRON INTERNETOWYCH: Bootstrap i Reactstrap są znane z responsywności. Zapewnia to, że aplikacja będzie dobrze wyglądać i działać na różnych urządzeniach i rozmiarach ekranu, co przekłada się na lepsze doświadczenie użytkownika.

WSPARCIE SPOŁECZNOŚCI: Reactstrap korzysta ze wsparcia i wkładu społeczności Reacta i Bootstrapa. Oznacza to, że można znaleźć wiele zasobów, dokumentacji i rozwiązań dla powszechnych problemów, co ułatwia efektywny rozwój oprogramowania.

ŁATWA INTEGRACJA Z ROZWIĄZANAMI OPARTYMI O REACT: Reactstrap, będąc specjalnie zaprojektowanym dla Reacta, doskonale integruje się z React. To zapewnia kompatybilność z innymi bibliotekami i narzędziami dla Reacta, ułatwiając budowę i utrzymanie aplikacji.

MODYFIKOWALNOŚĆ: Komponenty Reactstrap posiadają duże możliwości dostosowania ich wyglądu do konkretnych wymagań projektowych, nie tracąc jednocześnie zalet korzystania z biblioteki komponentów.

5.1.8 React Leaflet

React Leaflet jest biblioteką opakowującą funkcjonalności open-source'owej biblioteki javascriptowej Leaflet, która służy do wyświetlania interaktywnych map, w komponenty Reacta.

ZGODNOŚĆ Z REACT: Jako biblioteka React, React Leaflet doskonale wpisuje się w szerszy zbiór technologii opartych na React. Może być łatwo zintegrowany z innymi komponentami React, bibliotekami zarządzania stanem i frameworkami interfejsu użytkownika, zapewniając spójne doświadczenie programistyczne.

INTEGRACJA Z FUNKCJAMI LEAFLET: React Leaflet doskonale integruje się z biblioteką Leaflet, umożliwiając programistom korzystanie z bogatego zestawu funkcji dostarczanych przez tę bibliotekę.

ELASTYCZNOŚĆ I DOSTOSOWYWANIE: React Leaflet oferuje wysoki poziom elastyczności, umożliwiając dostosowywanie i rozszerzanie funkcji map. Można używać

różnych warstw mapy, dodawać niestandardowe markery i stosować style, aby stworzyć mapy zgodne z identyfikacją wizualną aplikacji.

OPTIMALIZACJA WYDAJNOŚCI: React Leaflet został zaprojektowany z myślą o wydajności. Efektywnie aktualizuje i ponownie renderuje tylko te komponenty, które są niezbędne po wprowadzeniu zmian, co sprawia, że mapa pozostaje responsywna nawet przy dynamicznych danych czy interakcjach użytkownika.

AKTYWNA KONSERWACJA I AKTUALIZACJE: React Leaflet jest aktywnie konserwowany i otrzymuje aktualizacje z rozwiązaniami błędów, wprowadzane są nowe funkcje oraz jest zapewniona zgodność z najnowszymi wersjami React i Leaflet. Dzięki temu programiści mogą korzystać z ciągłych ulepszeń i być na bieżąco z najlepszymi praktykami.

5.1.9 Leaflet GPX plugin

Wtyczka Leaflet GPX, oparta na pracy Pavla Shramova i jego wtyczek do Leaflet, umożliwia analizę i parsowanie ścieżki GPX w celu wyświetlenia jej jako warstwy mapy Leaflet.

Analizując dane GPX, może udostępniać informacje o zarejestrowanej trasie, w tym całkowity czas, czas ruchu, całkowity dystans, statystyki wysokości i tętno.

LATWOŚĆ INTEGRACJI: Plugin Leaflet GPX został stworzony z myślą o płynnej integracji z biblioteką Leaflet. Dzięki temu dodanie funkcjonalności GPX do aplikacji React korzystającej z React Leaflet nie nastręcza trudności. Plugin Leaflet GPX bazuje na Leaflecie zapewniając pełną kompatybilność z Leafletem.

WSPARCIE DLA PLIKÓW GPX: Głównym celem pluginu Leaflet GPX jest obsługa plików GPX, powszechnego formatu danych GPS. Pozwala wczytać trasy GPX i punkty na mapę Leaflet, co czyni go odpowiednim narzędziem do wizualizacji danych GPS.

MODYFIKOWALNOŚĆ: Plugin udostępnia opcje dostosowywania wyglądu tras GPX i punktów poprzez zmianę parametrów takich jak kolor, grubość linii czy ikony, dostosowując wygląd do wymagań projektowych.

OPEN SOURCE: Plugin pozwala na wgląd w kod źródłowy i w razie konieczności jego modyfikację.

5.1.10 JWT

JSON Web Token (JWT) to otwarty standard (RFC 7519), który definiuje kompaktowy i samowystarczalny sposób bezpiecznego przesyłania informacji między stronami jako obiekt JSON. Ta informacja może być zweryfikowana i uznana za wiarygodną dzięki cyfrowemu podpisowi.

UWIERZYTELNIENIE BEZSTANOWE: JWT jest bezstanowy, co oznacza, że wszystkie niezbędne informacje do zweryfikowania i autoryzacji użytkownika zawarte są w samym tokenie. Eliminuje to konieczność przechowywania danych sesji po stronie serwera, dzięki czemu jest bardziej skalowalny i odpowiedni dla bezstanowych interfejsów API i mikrousług.

KORZYŚCI WYDAJNOŚCIOWE: Ponieważ wszystkie niezbędne informacje są zawarte w samym tokenie, nie ma potrzeby częstego wykonywania przez serwer zapytań do bazy danych w celu weryfikacji danych uwierzytelniających użytkownika. Może to prowadzić do poprawy wydajności, zwłaszcza w scenariuszach z dużą liczbą żądań uwierzytelniających.

UWIERZYTELNIANIE MIĘDZYDOMENOWE: JWT obsługuje uwierzytelnianie międzydomenowe, umożliwiając aplikacji React bezpieczną komunikację z serwerem ASP.NET Core, nawet jeśli są hostowane na różnych domenach. Osiąga się to dzięki użyciu wspólnej tajemnicy lub par kluczy publicznych/prywatnych.

STANDARYZACJA I POWSZECHNA AKCEPTACJA: JWT to otwarty standard (RFC 7519), powszechnie akceptowany i obsługiwany w różnych językach programowania i platformach. Zapewnia to zgodność i łatwość integracji podczas tworzenia aplikacji opartych na różnych technologiach.

ZWIĘKSZONE BEZPIECZEŃSTWO: JWT może być podpisany i opcjonalnie zaszyfrowany, co wprowadza dodatkową warstwę bezpieczeństwa. Podpisanie pozwala serwerowi zweryfikować integralność tokenu, upewniając się, że nie został on sfalszowany. Szyfrowanie dodaje dodatkową ochronę, sprawiając, że zawartość tokenu jest nieczytelna dla osób bez klucza deszyfrującego.

ELASTYCZNOŚĆ W ZAKRESIE PRZEKAZYWANYCH TWIERDZEŃ: JWT może przenosić niestandardowe dane, pozwalając programistom na dołączanie dodatkowych informacji oprócz standardowej tożsamości użytkownika. Może to być przydatne do przekazywania ról, uprawnień lub innych istotnych danych.

UWIERZYTELNIENIE ZDECENTRALIZOWANA PO STRONIE KLIENTA: JWT

umożliwia uwierzytelnienie zdecentralizowane po stronie klienta. Po uwierzytelnieniu użytkownika, aplikacja React może przechowywać JWT lokalnie (np. w cookies) i dołączać go do kolejnych żądań. Ta decentralizacja zmniejsza zależność od zarządzania sesją po stronie serwera.

5.1.11 Azure Active Directory B2C

Usługa Azure Active Directory B2C dostarcza tożsamość firma-klient jako usługę. Klienci korzystają z preferowanych tożsamości społecznościowych, firmowych lub lokalnych kont, aby uzyskać dostęp do logowania jednokrotnego do aplikacji i interfejsów API.

SKALOWALNOŚĆ I WYDAJNOŚĆ: Usługi Azure AD B2C oferują wysoką skalowalność i niezawodność, co jest istotne dla obsługi żądań uwierzytelniania i autoryzacji w aplikacjach internetowych, w tym w SPA.

INTEGRACJA Z ASP.NET CORE: Usługi Azure AD B2C doskonale integruje się z różnymi usługami i rozwiązaniami Microsoftu, w tym z usługami Azure i z ASP.NET Core, integracja z Azure AD B2C upraszcza przepływy uwierzytelniania i autoryzacji.

WSPARCIE DLA OTWARTYCH STANDARDÓW I PROTOKOŁÓW: Usługi Azure AD B2C obsługują otwarte standardy i protokoły, takie jak OAuth 2.0, OpenID Connect i SAML, co sprawia, że jest kompatybilna z szeroką gamą aplikacji internetowych i mobilnych. Aplikacje SPA oparte na React często korzystają z OAuth 2.0 lub OpenID Connect do uwierzytelniania, a Azure AD B2C oferuje wszechstronne wsparcie dla tych protokołów.

ZARZĄDZANIE UŻYTKOWNIKAMI I DOSTOSOWYWANIE: Usługi Azure AD B2C umożliwiają definiowanie niestandardowych przepływów użytkownika, dostosowanych do konkretnych wymagań aplikacji. Obejmuje to funkcje takie jak rejestracja, logowanie, resetowanie hasła i zarządzanie profilem, które można dostosować do zgodności z marką i doświadczeniem użytkownika aplikacji SPA w React.

BEZPIECZEŃSTWO I ZGODNOŚĆ: Usługi Azure AD B2C zapewniają zaawansowane funkcje bezpieczeństwa, takie jak dwuskładnikowe uwierzytelnianie (2FA), polityki dostępu warunkowego i ochrona tożsamości, aby chronić konta użytkowników przed nieautoryzowanym dostępem i zagrożeniami bezpieczeństwa. Pomaga również organizacjom w spełnieniu wymagań związanych z przepisami regulacyjnymi, takimi jak RODO, HIPAA i ISO 27001.

PRODUKTYWNOŚĆ: Usługi Azure AD B2C mają obszerną dokumentację, zestawy SDK i narzędzia dla programistów w celu ułatwienia procesu integracji. Microsoft dostarcza biblioteki i przykłady dla różnych platform i języków programowania, w tym dla Reacta i ASP.NET Core, co ułatwia wdrożenie uwierzytelniania i autoryzacji w aplikacjach.

KOMPATYBILNOŚĆ MIĘDZYPLATFORMOWA: Azure AD B2C obsługuje uwierzytelnianie na różnych urządzeniach i platformach, w tym w przeglądarkach internetowych, urządzeniach mobilnych i aplikacjach na pulpit. Zapewnia to spójne i bezpieczne doświadczenie uwierzytelniania dla użytkowników korzystających z aplikacji SPA w React z różnych urządzeń i środowisk.

5.1.12 MSAL React

MSAL React, czyli Microsoft Authentication Library dla React, to biblioteka stworzona w celu ułatwienia integracji uwierzytelniania i autoryzacji w aplikacjach React przy współpracy z usługami Azure Active Directory B2C firmy Microsoft .

INTEGRACJA Z REACT: MSAL React jest specjalnie dostosowany do aplikacji React, dostarczając zestawu komponentów React i hooków, które ułatwiają bezproblemową integrację uwierzytelniania w React.

KOMPATYBILNOŚĆ Z USŁUGAMI AZURE: Trail Composer korzysta z usług Azure App Service i Azure SQL Database i użycie MSAL React jest naturalne w przypadku aplikacji obsługiwanej przez chmurę Azure, gdyż pozwala na możliwość dostępu do wszystkich usług tej chmury.

BEZPIECZNE UWIERZYTELNIANIE: MSAL React zapewnia bezpieczny i solidny mechanizm uwierzytelniania. Obsługuje standardowe protokoły OAuth 2.0 i OpenID Connect, co gwarantuje bezpieczne logowanie i kontrolę dostępu.

ZARZĄDZANIE TOKENAMI JWT: MSAL React zajmuje się pozyskiwaniem i zarządzaniem tokenami uwierzytelnienia i dostępu, które są kluczowe do bezpiecznej identyfikacji oraz wywołań interfejsów API, w tym do backendu ASP.NET Core i innych usług Azure.

DOKUMENTACJA I WSPARCIE SPOŁECZNOŚCI: Microsoft dostarcza wyczerpującą dokumentację, a wokół MSAL istnieje aktywna społeczność. Zapewnia to, że można znaleźć pomoc, zasoby i aktualizacje.

5.2 ARCHITEKTURA SYSTEMU

5.2.1 Założenia do projektowania architektury systemu:

- aplikacja będzie działać podobnie w środowisku rozwojowym i środowisku produkcyjnym i wykorzystywać te same usługi
- środowisko produkcyjne może być zapewnione przez dowolnego dostawcę usług hostingowych lub chmurowych
- aplikacja nie może wykorzystywać usług specyficznych dla danego dostawcy usług (z wyjątkiem usług uwierzytelnienia o ile te usługi mogą być wykorzystywane w środowisku dowolnego dostawcy).
- aplikacja jest rozwijana w zespole jednoosobowym więc nie jest konieczne zastosowanie rozwiązań zapewniających ciągłą integrację
- do przechowywania kodu aplikacji wykorzystuje się repozytorium Git hostowane na serwerze Github

5.2.2 Opis architektury aplikacji w środowisku rozwojowym

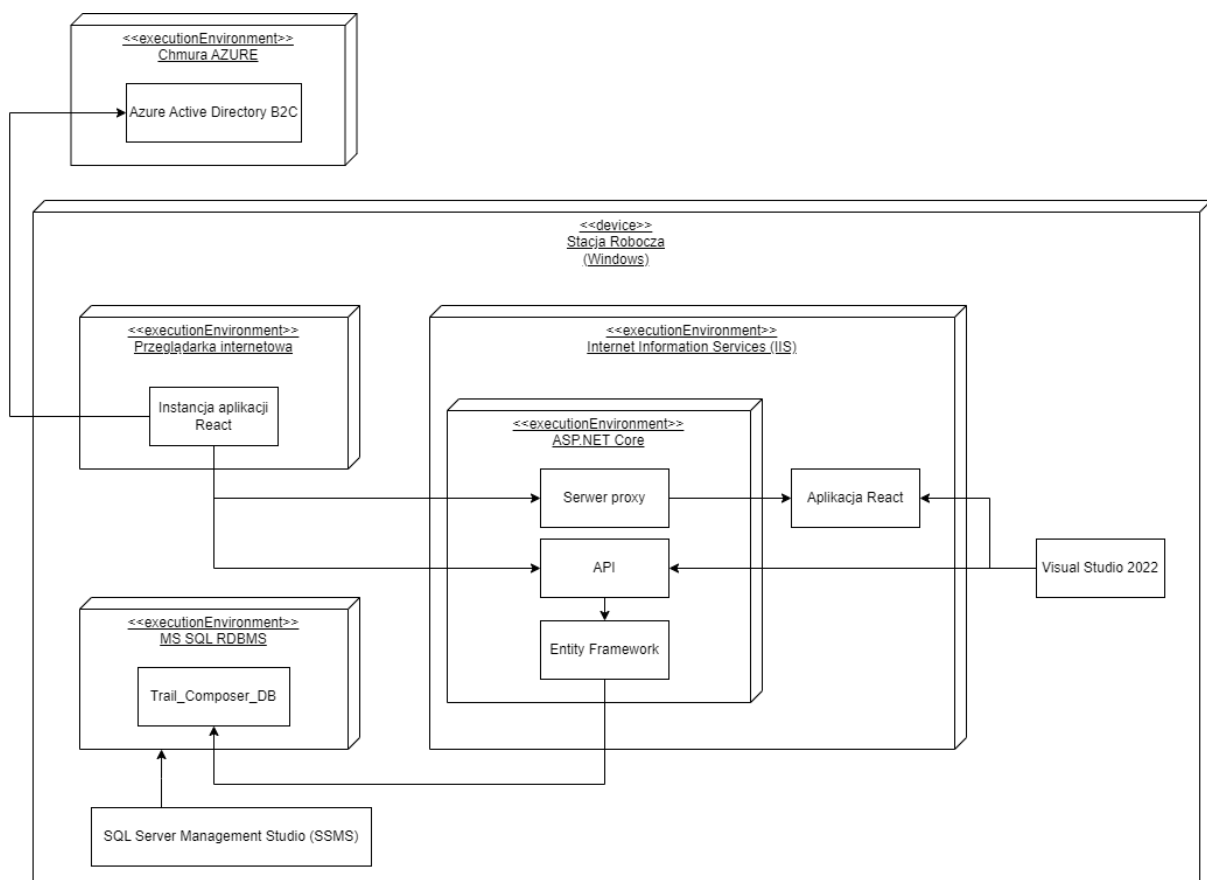


Diagram 1. Diagram architektury środowiska rozwoju aplikacji

Środowisko rozwojowe w całości (z wyjątkiem usług uwierzytelnienia i autoryzacji) będzie pracować na stacji roboczej z systemem Windows. Ponadto będzie wykorzystywać ono usługę Azure Active Directory B2C do logowania, uwierzytelniania, autoryzacji oraz zarządzania kontem użytkownika aplikacji.

Na stacji roboczej będzie uruchomiony serwer internetowy firmy Microsoft: Internet Information Services (IIS), który będzie hostować aplikację front-end wykonaną w technologii SPA z wykorzystaniem framework React oraz usługi backendowe API w technologii REST zbudowane na bazie ASP.NET Core, API wykorzystuje ORM framework firmy Microsoft: Entity Framework do integracji z bazą danych MS SQL Server.

Baza danych będzie zainstalowana lokalnie na stacji roboczej i będzie nią MS SQL Server.

Aplikacja będzie uruchamiana w przeglądarce Google Chrome na stacji roboczej.

Na tej samej stacji będzie pracować IDE firmy Microsoft Visual Studio 2022, które będzie służyło do rozwoju aplikacji oraz do przesyłania jej na serwer lokalny IIS.

Ponadto na stacji roboczej będzie wykorzystywane SQL Server Management Studio (SSMS) do zarządzania bazą danych MS SQL Server.

5.2.3 Opis architektury aplikacji w środowisku produkcyjnym

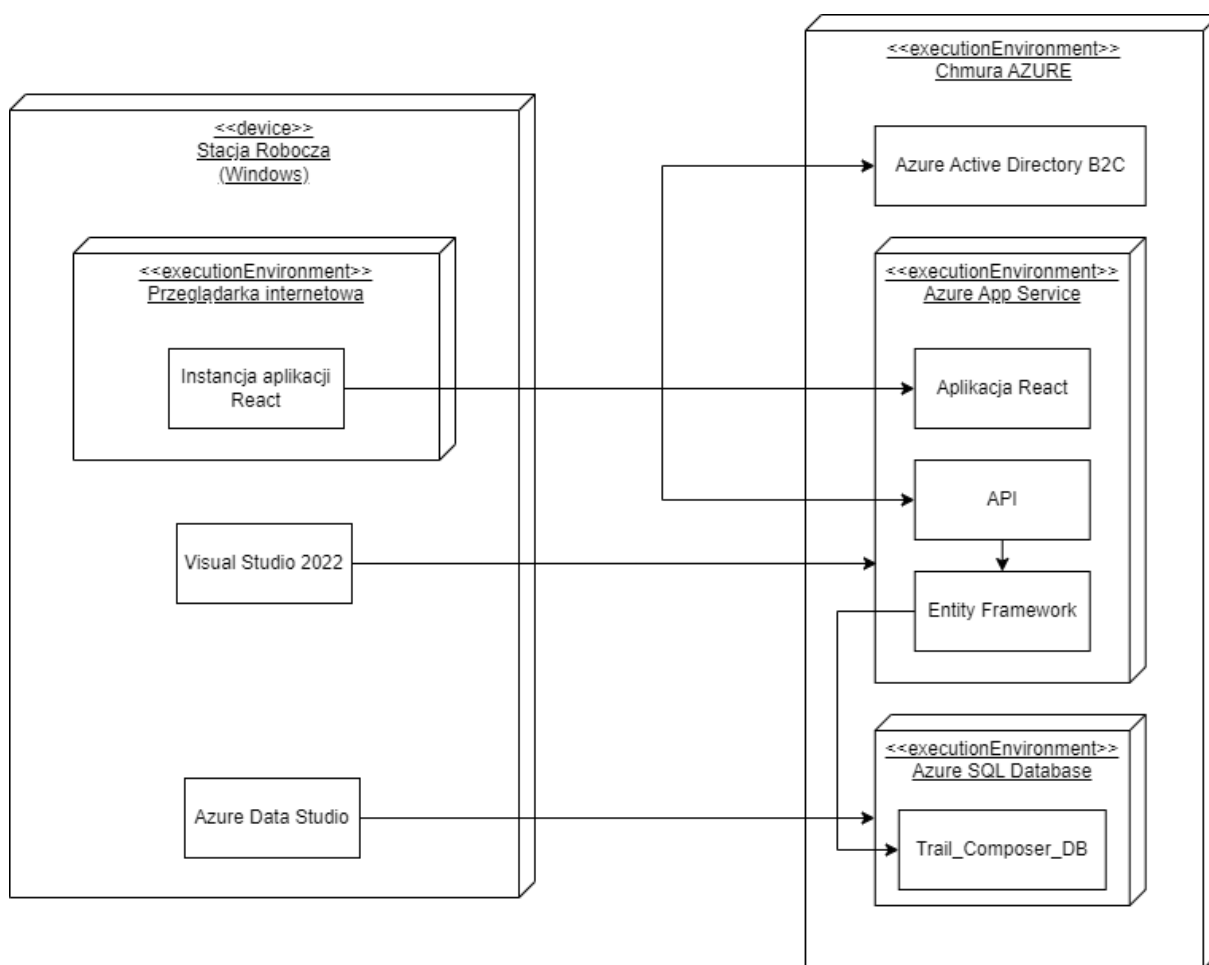


Diagram 2. Diagram architektury środowiska produkcyjnego aplikacji

Środowisko produkcyjne będzie wykorzystywać następujące usługi chmury AZURE:

- Usługę Azure Active Directory B2C do logowania, uwierzytelniania, autoryzacji oraz zarządzania kontem użytkownika aplikacji
- Usługę Azure App Service do hostowania aplikacji front-end wykonanej w technologii SPA z wykorzystaniem framework React oraz zapewnienia usług back-endowych przez API zbudowane na bazie ASP.NET Core w technologii REST, API wykorzystuje ORM framework firmy Microsoft: Entity Framework do integracji z bazą danych MS SQL Server
- Usługę Azure SQL Database zapewniającą dostęp do bazy danych MS SQL Server

Aplikacja będzie uruchamiana w przeglądarce Google Chrome na stacji roboczej pod systemem Windows.

Microsoft Visual Studio 2022 (i nowsze wersje) będzie służyło do rozwoju aplikacji oraz przesyłania jej do chmury Azure. Do zarządzania bazą danych w MS SQL Server dostępne w chmurze Azure będzie służyło Azure Data Studio.

5.2.4 Moduły aplikacji

5.2.4.1 Back-end:

- Moduł główny aplikacji
- Moduł obsługi baz danych (Entity Framework)
- Moduł uwierzytelnienia i autoryzacji (Microsoft Identity Web)
- Moduł zapisywania do logów w środowisku rozwojowym (Microsoft Extensions Logging)
- Moduł zapisywania do logów w środowisku produkcyjnym (Azure Application Insights)

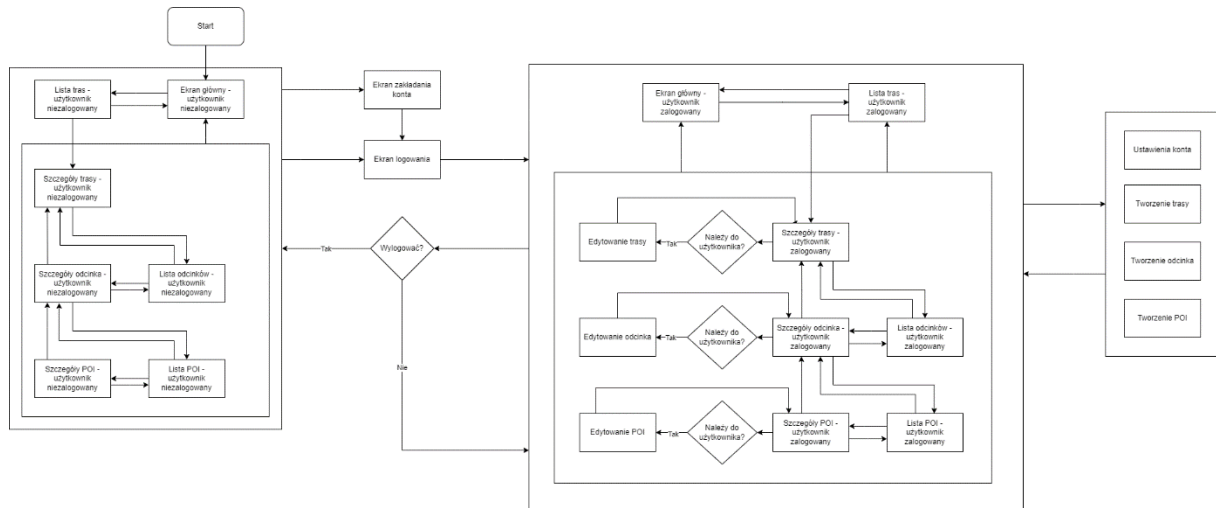
5.2.4.2 Front-end

- Moduł główny aplikacji
- Moduł routingu SPA (React Router)
- Moduł uniwersalnych komponentów (Reactstrap)
- Moduł obsługi map (React Leaflet)
- Moduł rysowania tras zawartych w plikach GPX (Leaflet GPX plugin)
- Moduł uwierzytelnienia i logowania (MSAL React)

5.3 PROJEKT INTERFEJSU

Prototypy ekranów znajdują się w załączniku Trail Composer mockup oraz pod linkiem

<https://www.figma.com/file/Trails-Composer-mockup>



Rysunek 7. Diagram przebiegów między ekranami

5.4 DIAGRAM ZWIĄZKÓW ENCJI Z OPISEM

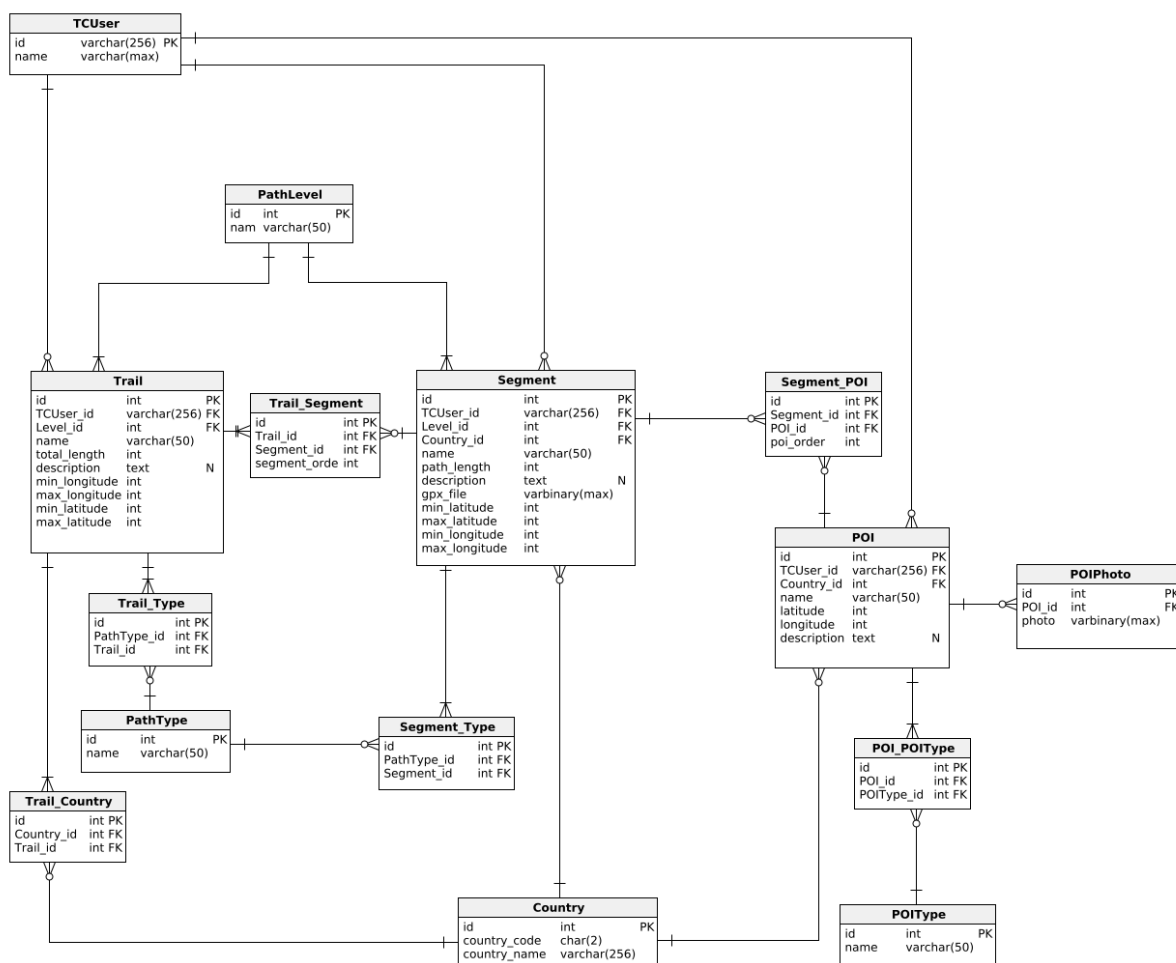


Diagram 3. Diagram związków encji

5.4.1 Tabela Trail

Tabela zawierające informacje o trasach.

Nazwa kolumny	Typ	Właściwości	Opis
Id	int	PK, IDENTITY (1,1)	identyfikator trasy
TCUser_id	varchar(256)	FK	identyfikator użytkownika będącego autorem trasy
Level_id	int	FK	identyfikator średniego poziomu trudności trasy
total_length	int		całkowita długość trasy w kilometrach
description	text	null	opis trasy
min_longitude	int		minimalna szerokość geograficzna opisująca bounding box (pudełko ograniczające) ograniczający trasę pomnożona razy 1 000 000
max_longitude	int		maksymalna szerokość geograficzna opisująca bounding box (pudełko ograniczające) ograniczający trasę pomnożona razy 1 000 000
min_latitude	int		minimalna długość geograficzna opisująca bounding box (pudełko ograniczające) ograniczający trasę pomnożona razy 1 000 000
max_latitude	int		maksymalna długość geograficzna opisująca bounding box (pudełko

			ograniczające) ograniczający trasę pomnożona razy 1 000 000
--	--	--	---

5.4.2 Tabela TCUser

Tabela zawierająca dane użytkowników Trail Composer korzystająca z usługi Azure AD B2C i ich bazy danych

Nazwa kolumny	Typ	Właściwości	Opis
id	varchar(256)	PK	unikalny identyfikator użytkownika w bazie danych Trail Composer
name	varchar(max)		nazwa wybrana przez użytkownika do

5.4.3 Tabela PathType

Tabela przechowująca informację o typach tras i odcinków

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator typu
name	varchar(50)		nazwa typu

5.4.4 Tabela PathLevel

Tabela zawierająca informacje o poziomie trudności trasy lub odcinka

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator poziomu
name	varchar(50)		nazwa poziomu

5.4.5 Tabela Trail_Type

Tabela reprezentująca relację wiele-do-wielu między trasami a ich rodzajami

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator relacji
PathType_id	int	FK	identyfikator typu trasy
Trail_id	int	FK	identyfikator trasy

5.4.6 Tabela Segment

Tabela zawierająca informację o odcinkach, z których składają się trasy

Nazwa kolumny	Type	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator odcinka
TCUser_id	varchar(256)	FK	identyfikator użytkownika będącego autorem odcinka
Level_id	int	FK	identyfikator poziomu trudności odcinka
Country_id	int	FK	identyfikator kraju przez który przebiega odcinek
name	varchar(50)		nazwa odcinka nadana przez użytkownika
path_length	int		długość odcinka w metrach
description	text	null	opis odcinka

gpx_file	varbinary(max)		plik gpx z przebiegiem odcinka
min_latitude	int		minimalna szerokość geograficzna opisująca bounding box (pudełko ograniczające) ograniczający odcinek pomnożona razy 1 000 000
max_latitude	int		maksymalna szerokość geograficzna opisująca bounding box (pudełko ograniczające) ograniczający odcinek pomnożona razy 1 000 000
min_longitude	int		minimalna długość geograficzna opisująca bounding box (pudełko ograniczające) ograniczający odcinek pomnożona razy 1 000 000
max_longitude	int		maksymalna długość geograficzna opisująca bounding box (pudełko ograniczające) ograniczający odcinek pomnożona razy 1 000 000

5.4.7 Tabela Trail_Segment

Tabela reprezentująca relację wiele-do-wielu między trasami a odcinkami, z których się składają

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY(1,1)	identyfikator relacji
Country_id	int	FK	identyfikator kraju
Trail_id	int	FK	identyfikator trasy
segment_order	int		Kolejność odcinków tworzących trasę

5.4.8 Tabela POI

Tabela przechowująca dane o Point of Interests (POI)

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator POI
TCUser_id	varchar(256)	FK	identyfikator użytkownika, który jest autorem POI
Country_id	int	FK	identyfikator kraju, w którym leży POI
name	varchar(50)		nazwa POI
latitude	decimal(10,6)		szerokość geograficzna, na której leży POI
longitude	decimal(10,6)		długość geograficzna, na której leży POI
description	text	null	opis POI

5.4.9 Tabela Segment_POI

Tabela reprezentująca relację wiele-do-wielu między odcinkami a POI, przez które przebiegają

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator relacji
Segment_id	int	FK	identyfikator odcinka
POI_id	int	FK	identyfikator POI
poi_order	int		Kolejność POI należących do odcinka

5.4.10 Tabela Segment_Type

Tabela reprezentująca relację wiele-do-wielu między odcinkami a ich rodzajami

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator relacji
PathType_id	int	FK	identyfikator typu odcinka
Segment_id	int	FK	identyfikator odcinka
poi_order	int		Kolejność POI należących do odcinka

5.4.11 Tabela POIType

Tabela przechowująca informacje o typach POI

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY(1,1)	identyfikator typu
name	varchar(50)		Nazwa typu

5.4.12 Tabela POI_POIType

Tabela reprezentująca relację wiele-do-wielu między POI a ich rodzajami

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator relacji
POI_id	int	FK	identyfikator POI
POIType_id	int	FK	identyfikator typu POI

5.4.13 Tabela Country

Tabela przechowująca informacje o krajach

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator kraju
country_code	char(2)		dwuznakowy kod kraju zgodny z normą ISO 3166-1 alpha-2
country_name	char(255)		nazwa kraju w powszechnym użyciu

5.4.14 Tabela Trail_Country

Tabela reprezentująca relację wiele-do-wielu między trasami a krajami, przez które przebiegają

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator relacji
Country_id	int	FK	identyfikator kraju
Trail_id	int	FK	identyfikator trasy

5.4.15 Tabela POIPhoto

Tabela zawierająca informację o zdjęciach przypisanych do POI

Nazwa kolumny	Typ	Właściwości	Opis
id	int	PK, IDENTITY (1,1)	identyfikator zdjęcia
POI_id	int	FK	identyfikator POI, do którego należy zdjęcie
photo	varbinary(max)		plik binarny ze zdjęciem w formacie jpg

5.5 WYMAGANIA SPRZĘTOWE

5.5.1 Wymagania sprzętowe dla środowiska rozwojowego

5.5.1.1 Minimalne Wymagania:

Procesor (CPU)	Intel Core i5 lub AMD Ryzen 5
RAM	8 GB
Pamięć masowa	256 GB SSD
Karta graficzna	Zintegrowana karta graficzna
System operacyjny	Windows 10 lub nowszy
Liczba dodatkowych monitorów o przekątnej minimum 24"	1
Połączenie z Internetem	tak

5.5.1.2 Zalecane Wymagania:

Procesor (CPU)	Intel Core i7 lub AMD Ryzen 7
RAM	16 GB lub więcej
Pamięć masowa	512 GB SSD lub więcej
Karta graficzna	Zintegrowana karta graficzna
System operacyjny	Windows 10 Pro lub nowszy
Liczba dodatkowych monitorów o przekątnej minimum 24"	2-3
Połączenie z Internetem	tak

Wymagania sprzętowe dla środowiska produkcyjnego

5.5.1.3 Wymagania na etapie rozwoju aplikacji i obrony pracy inżynierskiej:

- Azure Active Directory B2C Service Plan

Poziom cenowy (Pricing Tier)	dzierżawa usługi połączona z subskrypcją płatną Pay-As-You-Go do Azure, opłaty będą automatycznie naliczane przy użyciu modelu rozliczeń opartego na ilości MAU (Miesięcznych Aktywnych Użytkowników). Pierwsze 50 000 jednostek MAU miesięcznie jest bezpłatne
Ograniczenia czasowe oferty (time limited offer)	nie ma (no time limit)
Regiony (Regions)	Polska Centralna (Central Poland)

- Azure App Service Plan

Poziom cenowy (Pricing Tier)	darmowa (free)
Ograniczenia czasowe oferty (time limited offer)	nie ma (no time limit)
Skalowanie w górę (Scaling Up)	nie
Skalowanie w dół (Scaling Down)	nie
Automatyczne skalowanie (Dynamic Scaling/Auto Scaling)	nie
Ograniczenia na max/min liczbę instancji (Scaling Limit)	w wersji darmowej jest tylko jedna instancja na subskrypcję
Regiony (Regions)	Polska Centralna (Central Poland)
System operacyjny (Operating System)	Windows
Rozmiar instancji (Instance Size)	F1 (Rdzenie: 1, RAM: 1GB, Pamięć masowa: 1GB)
Wysoka dostępność (High Availability)	rozproszenie po regionach nie jest wymagane
Środowiska uruchomieniowe (Deployment Slots)	tylko jedno - produkcyjne

- Azure SQL Database Plan

Poziom cenowy (Pricing Tier)	darmowa (free)
Ograniczenia czasowe oferty (time limited offer)	nie ma (no time limit)
Automatyczne skalowanie (Dynamic Scaling/Auto Scaling)	nie
Ograniczenia na max/min liczbę instancji (Scaling Limit)	w wersji darmowej jest tylko jedna instancja na subskrypcję
Automatyczne zatrzymanie bazy po przekroczeniu limitu miesięcznego i automatyczne uruchomienie w następnym miesiącu (database auto-pause)	tak (yes)
Regiony (Regions)	Polska Centralna (Central Poland)
System operacyjny (Operating System)	Windows
Wysoka dostępność (High Availability)	rozproszenie po regionach nie jest wymagane
Dostępne zasoby	
czas obliczeniowy wirtualnego rdzenia (vCore seconds)	100 000 sekund miesięcznie
pamięć masowa (mass storage)	32 GB na dane + 32 GB na kopię zapasową (backup)

5.5.1.4 Wymagania w pozostałych etapach wykorzystania aplikacji:

Wymagania zależą od modelu biznesowego dla aplikacji: czy będzie to wykorzystanie komercyjne czy “pro publico bono” oraz możliwości pozyskania źródeł finansowania. W przypadku źródeł finansowania uzależnionych od ruchu w aplikacji np. reklamy model skalowania dynamicznego może być uzasadniony. W przypadku finansowania na stałym poziomie kosztowym, konfiguracja z ograniczeniami na maksymalną liczbę instancji i limitami ruchu powinna być bezpieczniejsza. Generalnie byłaby potrzebna większa ilość środowisk uruchomieniowych: środowisko do testów programistycznych, środowisko do testów akceptacyjnych, środowisko produkcyjne. Ponadto warto by było wdrożyć AZURE CDN dla React SPA aby mieć niższe koszty oraz sprawniejszą obsługę użytkowników rozproszonych po świecie. Zaletą wdrożenia w chmurze będzie możliwość dostosowania wymagań sprzętowych do zmieniających się modeli biznesowych i ruchu użytkowników bez konieczności ponoszenia kosztów inwestycyjnych.

6 WYNIKI TESTÓW I DYSKUSJA

6.1 TESTY FUNKCJONALNE

Ostateczne testy funkcjonalne zostały przeprowadzone w ramach sporządzania instrukcji obsługi. Wszystkie przypadki użycia działały zgodnie z założeniami za wyjątkiem UC 2.6 „Wyświetl szczegóły POI” i UC 2.7 „Pobierz plik GPX wybranego odcinka”:

- UC 2.6: elementy listy POI nie przenosiły użytkownika do szczegółów wybranego POI z powodu nie przekazania odpowiedniej funkcji do komponentu tabeli. Błąd został naprawiony.
- UC 2.7: Przycisk „Pobierz GPX” był widoczny i pozwalał na pobranie pliku GPX użytkownikowi niezalogowanemu co jest niezgodne z założeniami. Błąd został naprawiony.

6.2 TEST NIEZAWODNOŚCIOWE

7 BIBLIOGRAFIA

- [1] „React Router tutorial,” [Online]. Available: <https://reactrouter.com/en/main/start/tutorial>.
- [2] „React Reference Documentation,” [Online]. Available: <https://react.dev/reference/react>.
- [3] „Reactstrap Reference Documentation,” [Online]. Available: <https://reactstrap.github.io/>.
- [4] „Kod źródłowy Reactstrap,” [Online]. Available: <https://github.com/reactstrap/reactstrap>.
- [5] „React Leaflet Documentation,” [Online]. Available: <https://react-leaflet.js.org/docs/start-introduction/>.

- [6] „GPX plugin for Leaflet,” [Online]. Available: <https://github.com/mpetazzoni/leaflet-gpx#readme>.
- [7] M. B. Jones, J. Bradley i N. Sakimura, „Internet Standards Track document. JSON Web Token (JWT),” 05 2015. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7519>.
- [8] „Auth0 by Okta, Debugger. Libraries. Introduction,” [Online]. Available: <https://jwt.io/>.
- „Co to jest usługa Azure Active Directory B2C?,” 22 09 2023. [Online]. Available: <https://learn.microsoft.com/pl-pl/azure/active-directory-b2c/overview#identity-providers>.
- [9] u[9] <https://learn.microsoft.com/pl-pl/azure/active-directory-b2c/overview#identity-providers>.
- [10] „Szybki start: konfigurowanie logowania dla aplikacji jednostronicowej przy użyciu usługi Azure Active Directory B2C,” 23 05 2023. [Online]. Available: <https://learn.microsoft.com/pl-pl/azure/active-directory-b2c/quickstart-single-page-app>.
- [11] „Enable authentication in your own web app by using Azure AD B2C,” 11 01 2024. [Online]. Available: <https://learn.microsoft.com/en-us/azure/active-directory-b2c/enable-authentication-web-application?tabs=visual-studio>.
- [12] „Model rozliczeń dla usługi Azure Active Directory B2C,” 11 01 2024. [Online]. Available: <https://learn.microsoft.com/pl-pl/azure/active-directory-b2c/billing>.
- [13] Microsoft Corporation, „Microsoft Authentication Library for React (msal-react),” [Online]. Available: <https://github.com/AzureAD/microsoft-authentication-library-for-js/tree/dev/lib/msal-react>.
- [14] „Azure Active Directory B2C: dostępność regionów i miejsce przechowywania danych,” 11 01 2024. [Online]. Available: <https://learn.microsoft.com/pl-pl/azure/active-directory-b2c/data-residency>.
- [15] „biblioteki uwierzytelniania Platforma tożsamości Microsoft,” 24 10 2023. [Online]. Available: <https://learn.microsoft.com/pl-pl/entra/identity-platform/reference-v2-libraries>.

- [16] „Samouczek: rejestrowanie aplikacji jednostronicowej przy użyciu Platforma tożsamości Microsoft,” 14 12 2023. [Online]. Available: <https://learn.microsoft.com/pl-pl/entra/identity-platform/tutorial-single-page-app-react-register-app>.
- [17] „Microsoft Docs, Managed identities for Azure resources,” [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/>.
- [18] „Microsoft Azure, Azure SQL Database documentation,” [Online]. Available: <https://docs.microsoft.com/en-us/azure/azure-sql/database/>.
- [19] „Microsoft Docs, Azure SQL Database service tiers and performance levels,” [Online]. Available: <https://docs.microsoft.com/en-us/azure/azure-sql/database/service-tier-purchasing-models>.
- [20] „Microsoft Azure, Pricing - SQL Database,” [Online]. Available: <https://azure.microsoft.com/en-us/pricing/details/sql-database/>.
- [21] „App Service documentation,” [Online]. Available: <https://learn.microsoft.com/en-us/azure/app-service/>.
- [22] „Microsoft Docs, App Service plans in Azure,” [Online]. Available: <https://docs.microsoft.com/en-us/azure/app-service/overview-hosting-plans>.
- [23] „C# documentation,” [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/>.
- [24] „ASP.NET documentation,” [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>.
- [25] „Entity Framework documentation hub,” [Online]. Available: <https://learn.microsoft.com/en-us/ef/>.
- [26] „Microsoft SQL documentation,” [Online]. Available: <https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16>.

8 ZAŁĄCZNIKI

- Diagram architektury środowiska rozwoju aplikacji
- Diagram architektury środowiska produkcyjnego aplikacji
- Trail Composer mockup
- Diagram przejść między ekranami
- Diagram encji