

E - książeczka GOT PTTK

Projektowanie oprogramowania



E - GOT PTTK

Skład zespołu:

Agnieszka Kłobus
Aleksandra Stecka
(zespół 8)

Prowadzący:

dr inż. Zbigniew Szpunar

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Spis treści:

Wyniki etapu 1: Wizja, słownik, model domenowy, reguły biznesowe

Wizja

- Wprowadzenie
- Sformułowanie problemu
- Opis pozycji produktu
- Opis udziałowców i użytkowników
- Opis produktu

Słownik pojęć

Model domenowy

Reguły biznesowe

Wyniki etapu 2: Specyfikacja wymagań, diagram przypadków użycia, model informacyjny, prototyp interfejsu, specyfikacja przypadków użycia

Specyfikacja wymagań

Model przypadków użycia i prototypy interfejsów

Diagram przypadków użycia

Specyfikacja przypadków użycia - Agnieszka Kłobus

Specyfikacja przypadków użycia - Aleksandra Stecka

Prototyp interfejsu

Zamieszczenie dowodu

Zapisanie trasy w książeczkę

Edycja odcinka w spisie

Potwierdzenie trasy

Model informacyjny

Wyniki etapu 3: Architektura systemu, projekt bazy danych, realizacja przypadków użycia

Architektura oprogramowania

Architektura logiczna

Architektura fizyczna

Model danych

Realizacja przypadków użycia

Projekt bazy danych

Skrypt SQL DDL

Wyniki etapu 4: Implementacja, testy jednostkowe, ocena jakości

Uzupełnienie Etapu 3

Zamieszczenie dowodu – Agnieszka Kłobus

Zapisanie trasy w książeczkę – Agnieszka Kłobus

Edycja odcinka w spisie – Aleksandra Stecka

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Potwierdzenie trasy – Aleksandra Stecka
Implementacja
Struktura kodu
Wzorce projektowe
Repozytorium na serwisie Github
Testowanie
test repozytorium OfficialPointRepository
test metody pointsNotInTheSameRange w klasie Utils
test repozytorium LeaderRepository
test metody formatDate w klasie Utils
Przypadki testowe dla PU
Edycja odcinka
Zapisanie trasy w książeczce
Dodanie dowodu
Potwierdzenie trasy
Automatyzacja testów funkcjonalnych
Edycja odcinka
Zapisanie trasy w książeczce
Dodanie dowodu
Potwierdzenie trasy

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Wyniki etapu 1: Wizja, słownik, model domenowy, reguły biznesowe

E - książeczka GOT PTTK

Projektowanie oprogramowania



Skład zespołu:
Agnieszka Kłobus
Aleksandra Stecka
(zespół 8)

Prowadzący:
dr inż. Zbigniew Szpunar

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Wizja

Wprowadzenie

Górską Odznaką Turystyczną Polskiego Towarzystwa Turystyczno – Krajoznawczego jest odznaką ustanowioną w celu promowania turystyki górskiej. Zdobycie tej odznaki wiąże się więc z uprawianiem turystyki górskiej.

Najpopularniejsze rodzaje GOT PTTK to:

- popularna – o jednym stopniu,
- mała – o trzech stopniach, brązowym, srebrnym i złotym.

Za wycieczki, na które wybierają się turyści otrzymują oni punkty, na podstawie których mogą ubiegać się o odznakę. Aby zdobyć odznakę popularną należy zdobyć 60 punktów. Na odznakę małą o stopniu brązowym należy zdobyć 120 punktów, o stopniu srebrnym 360 punktów, a o stopniu złotym 720 punktów. Jeżeli trasa wycieczki jest ujęta w spisie tras punktowanych do GOT PTTK należy stosować podaną w nim punktację. W przeciwnym wypadku przyznaje się jeden punkt za każdy przebyty kilometr oraz jeden punkt za każde pokonane 100 metrów różnicy poziomów przy podejściach.

Trasy wycieczek turystów składają się z odcinków – jednego lub wielu. Każdy odcinek trasy wycieczki łączy pewne dwa punkty geograficzne. Spis tras punktowanych do GOT PTTK jest opracowywany przez Centralny Ośrodek Turystyki Górskiej PTTK i można znaleźć w nim odcinki tras wycieczek, które mają przypisaną oficjalną punktację. Pomiędzy dwoma punktami geograficznymi może być wytyczone wiele odcinków tras, które mogą mieć różną punktację. Niektóre z odcinków tras są dwukierunkowe – za przejście danego odcinka trasy w jedną stronę można otrzymać pewną liczbę punktów, a w drugą stronę inną liczbę punktów. Inne odcinki tras są jednokierunkowe, co oznacza, że posiadają przypisaną punktację tylko w jednym kierunku.

Jeśli turysta wybiera się na wycieczkę o długiej trasie, składającej się z wielu odcinków, otrzymuje punkty za każdy z odcinków tej trasy. Maksymalna liczba punktów, które turysta może otrzymać jednego dnia to 50 punktów. Oprócz tego, nie zalicza się punktów uzyskanych na wycieczkach przebytych tą samą trasą i w tym samym kierunku przy zdobywaniu tego samego stopnia GOT PTTK.

Może wystąpić sytuacja, w której jakiś odcinek trasy jest czasowo wyłączony – np. został zablokowany przez drzewo lub osuwisko skalne. Czasowego wyłączenia odcinków trasy nie uwzględniono w projekcie. Za wszystkie odcinki, które turysta wpisał do książeczki w momencie, w którym były zablokowane, nie dostanie on punktów ponieważ książeczki podlegają weryfikacji i sprawdzaniu przez osoby współpracujące z GOT PTTK.

E-ksiązeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Występują również sytuacje, kiedy jakiś odcinek trasy został całkowicie zlikwidowany – np. teren, przez który przebiegał, został zalany. Wtedy turyści, którzy nie przebyli jeszcze tego odcinka, nie mogą już tego zrobić, ale nie wolno utracić informacji o turystach, którzy przebyli go przed likwidacją.

Dokumentem uprawniającym do nabycia i noszenia odznak GOT PTTK jest książeczka GOT PTTK. Książeczkę należy wypełniać samodzielnie, a następnie podać ją weryfikacji. Istotnym elementem książeczki jest załączenie dowodu przebycia wpisanej trasy - w postaci zdjęć pieczątek z nazwami miejscowości, schronisk PTTK czy charakterystycznych elementów krajobrazu. Dostępne jest także załączenie dowodu przebycia trasy w postaci zrzutu ekranu pochodzącego z dowolnej aplikacji śledzącej lokalizację użytkownika. Dowód przebycia trasy nie jest konieczny jeśli przodownik turystyki górskiej PTTK uczestniczył w danej wycieczce i potwierdzi ten fakt podpisem oraz numerem legitymacji przodownickiej oraz, jeśli to możliwe, imienną pieczęcią.

Trasy wpisane do książeczki GOT PTTK muszą zostać potwierdzone przez przodownika turystyki górskiej PTTK. Przodownik ma uprawnienia w danej grupie górskiej i określa, czy turysta faktycznie przebył wpisaną trasę na podstawie załączonych dowodów czy daty wycieczki. Wszystkie trasy wpisane do książeczki GOT PTTK muszą być potwierdzone przez przodownika.

Książeczki osób ubiegających się o GOT PTTK popularną i małą weryfikują Terenowe Referaty Weryfikacyjne GOT PTTK lub Centralny Referat Weryfikacyjny GOT PTTK.

Celem projektu jest zaoferowanie alternatywy dla papierowej książeczki GOT PTTK w postaci systemu elektronicznego. System będzie zapewniał możliwość wprowadzania, modyfikacji oraz usuwania danych dotyczących odcinków tras punktowanych do GOT PTTK przez użytkowników do tego uprawnionych, a także wyszukiwania i prezentacji odcinków tras przez wszystkich użytkowników. Oprócz tego, system będzie umożliwiał dokumentowanie tras wycieczek odbytych w ramach zdobywania kolejnych rodzajów i stopni GOT PTTK.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Sformułowanie problemu

Problem	Brak pełnej bazy odcinków tras wliczanych do GOT PTTK umożliwiającej wyszukiwanie i filtrowanie wyników oraz brak oprogramowania pozwalającego turyście przechowywać informacje o przebytych trasach w formie elektronicznej.
Dotyczy	Osób odpowiedzialnych za GOT PTTK i trasy wycieczek, przodowników turystyki górskiej, turystów posiadających książeczkę GOT PTTK oraz innych zainteresowanych zdobyciem informacji na temat GOT PTTK.
Wpływ problemu	Brak możliwości wyszukiwania i filtrowania odcinków Utrudniony proces potwierdzenia zdobycia kolejnych odznak Długi czas oczekiwania na potwierdzenie przebycia trasy Podniesiony koszt zdobycia odznaki GOT PTTK związany z ponoszeniem kosztów wysyłki pocztowej Ograniczony dostęp do informacji na temat tras
Pomyślne rozwiązanie	Zwiększy zainteresowanie zdobywaniem odznak GOT PTTK poprzez ułatwienie tego procesu oraz uproszczenie wyszukiwania informacji na temat tras Zmniejszy czasu oczekiwania na potwierdzenie przebycia trasy oraz weryfikację książeczkę

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Opis pozycji produktu

Dla	Komisji Turystyki Górskiej Polskiego Towarzystwa Turystyczno-Krajoznawczego
Który/Która/Którzy	Chce usprawnić proces potwierdzania przebycia trasy oraz weryfikacji książeczków, ułatwić wyszukiwanie informacji o odcinkach oraz zwiększyć zainteresowanie turystyką górską
E - GOT PTTK	Oprogramowanie Bazodanowe dla Górskiej Odznaki Turystycznej PTTK (GOT PTTK)
Który	Umożliwia łatwy dostęp do danych oraz zarządzanie nimi
Inaczej niż	Aktualnie istniejące listy tras oraz tradycyjnie używane książeczkki
Nasz produkt	Usprawnia dostęp do danych oraz zarządzanie nimi poprzez zastąpienie dokumentów papierowych systemem elektronicznym.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Opis udziałowców i użytkowników

Podsumowanie udziałowców

Nazwa	Opis	Zakres odpowiedzialności
Polskie Towarzystwo Turystyczno Krajoznawcze	Organizacja odpowiedzialna za akcję GOT PTTK ustalającą trasy oraz ich punktację.	Monitoruje proces powstawania oprogramowania i akceptuje projekt Zapewnia fundusze na realizację projektu Zapewnia utrzymanie działającego oprogramowania Zarządza danymi przechowywanymi w systemie
Zespół projektantów	Osoby wykonujące projekt systemu dla PTTK	Projektuje system Implementuje system

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Podsumowanie użytkowników

Nazwa	Opis	Zakres odpowiedzialności
Członek Komisji Turystyki Górskiej Zarządu Głównego PTTK	Członek komisji sprawującej nadzór nad GOT PTTK	Przeglądanie, edycja, dodawanie, usuwanie i aktualizowanie danych dotyczących odcinków tras Zarządzanie systemem
Pracownik Centralnego Ośrodka Turystyki Górskiej PTTK	Osoba zatrudniona w Centralnym Ośrodku Turystyki Górskiej PTTK	Nadawanie uprawnień przodownikom turystyki górskiej
Pracownik Referatu Weryfikacyjnego GOT PTTK	Osoba zatrudniona w Centralnym Referacie Weryfikacyjnym lub Terenowym Referacie Weryfikacyjnym	Przyznanie odznaki turystom na podstawie weryfikacji książeczek GOT PTTK
Przodownik Turystyki Górskiej	Członek społecznej kadry programowej PTTK propagujący GOT PTTK	Potwierdzenia odbytych przez turystę wycieczek górskich w elektronicznej książeczce GOT PTTK turysty
Turysta	Osoba uprawiająca turystykę górską, posiadająca książeczkę GOT PTTK	Przeglądanie i wyszukiwanie tras w bazie Rejestrowanie odbytych wycieczek Wysyłanie przebytych tras do potwierdzenia Wysyłanie książeczek do weryfikacji
Gość	Osoba zainteresowana turystyką górską	Przeglądanie i wyszukiwanie tras w bazie

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Opis produktu

Potrzeby i cechy

Potrzeba	Priorytet	Cechy	Planowane wydanie
Logowanie	Wysoki	Każdy użytkownik posiada email oraz hasło za pomocą których loguje się do systemu	Wersja I
Autoryzacja	Wysoki	Różni użytkownicy mają uprawnienia do wykonywania różnych akcji w systemie	Wersja I
Przechowywanie informacji o odcinkach	Wysoki	Uprawnieni użytkownicy mają możliwość dodawania, edycji i usuwania danych	Wersja I
Wyszukiwanie i filtrowanie odcinków	Wysoki	Użytkownicy mają możliwość wyszukiwać odcinki spełniające podane kryteria	Wersja I
Zapisywanie przebytej trasy na koncie turysty	Średni	Automatyczne zliczanie punktów za przebycie punktowanych odcinków Turysta ma możliwość wprowadzić własną punktację za odcinki spoza spisu tras punktowanych do GOT PTTK	Wersja II
Przypisanie elektronicznej książeczki GOT do turysty	Średni	Turysta ma możliwość rejestracji odbytych wycieczek Sumowane są punkty turysty do odznaki GOT	Wersja II
Potwierdzenie trasy przebytej	Średni	Przodownik turystyki górskiej PTTK zatwierdza	Wersja II

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

przez turystę		trasy turystów	
Przyznanie odznaki turystom	Niski	Terenowy Referat Weryfikacyjny GOT PTTK lub Centralny Referat Weryfikacyjny GOT PTTK weryfikuje spełnienie przez turystę wymogów na odpowiedni rodzaj lub stopień odznaki GOT PTTK i przyznaje odznakę.	Wersja III

Inne wymagania produktowe

Wymaganie	Priorytet	Planowane wydanie
Polska wersja językowa	Wysoki	Wersja I
Zabezpieczenie kopii zapasowej	Wysoki	Wersja I
Możliwość uruchomienia za pomocą aplikacji mobilnej	Wysoki	Wersja I
Dostępność dokumentacji i instrukcji obsługi dla użytkowników	Wysoki	Wersja I
Plynna obsługa dużej ilości użytkowników	Średni	Wersja II
Wsparcie czytników dla osób słabowidzących	Niski	Wersja III

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Słownik pojęć

Termin	Synonimy	Definicja terminu
Centralny Ośrodek Turystyki Górskiej PTTK	COTG PTTK	Jednostka specjalistyczna PTTK, której pracownicy nadają uprawnienia przodownikom turystyki górskiej.
Centralny Referat Weryfikacyjny GOT PTTK	CRW GOT PTTK	Centralna jednostka specjalistyczna PTTK, weryfikująca książeczki GOT PTTK turystów.
Członek Komisji Turystyki Górskiej Zarządu Głównego PTTK	Członek KTG ZG PTTK	Członek komisji sprawującej nadzór nad GOT PTTK, który dodaje nowe i edytuje istniejące odcinki w spisie odcinków punktowanych do GOT PTTK. Posiada indywidualne konto w systemie i jest do niego zalogowany.
Dowód		Dowód potwierdzający przebycie odcinka. Jest zdjęciem pieczętki z nazwą miejscowości, tablicy z nazwą miejscowości, schroniska, charakterystycznych elementów krajobrazu lub zrzutem ekranu z aplikacji śledzącej lokalizację.
Gość		Osoba zainteresowana turystyką górską, która nie utworzyła jeszcze konta w systemie lub nie jest zalogowana na swoje konto.
Górska Odznaka Turystyczna mała	GOT mała, odznaka mała	Jeden z rodzajów Górskiej Odznaki Turystycznej, mający trzy stopnie: brązowy, srebrny i złoty.
Górska Odznaka Turystyczna mała brązowa	odznaka mała brązowa, GOT mała brązowa	Rodzaj i stopień GOT, który przyznawany jest turyście za zdobycie 120 punktów.
Górska Odznaka Turystyczna mała srebrna	odznaka mała srebrna, GOT mała srebrna	Rodzaj i stopień GOT, który przyznawany jest turyście za zdobycie 360 punktów.
Górska Odznaka Turystyczna mała złota	odznaka mała złota, GOT mała złota	Rodzaj i stopień GOT, który przyznawany jest turyście za zdobycie 720 punktów.
Górska Odznaka Turystyczna Polskiego Towarzystwa	Górska Odznaka Turystyczna, GOT PTTK, odznaka GOT	Odznaka ustanowiona w 1935 przez Polskie Towarzystwo Tatrzańskie, promująca turystykę górską. Ma poszczególne rodzaje i stopnie. Jest

E-ksiązeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Turystyczno-Krajoznawczego	PTTK, odznaka GOT, GOT, odznaka	przyznawana w konkretnym dniu.
Górsko Odznaka Turystyczna popularna	odznaka popularna, GOT popularna	Rodzaj GOT, który przyznawany jest turyście za zdobycie 60 punktów. Posiada jeden stopień.
Grupa górska		Zbiór pasm górskich leżących w bliskim położeniu od siebie. Jest identyfikowana przez nazwę i znajduje się w konkretnym kraju. Przodownik ma uprawnienia w ramach grupy górskiej.
Komisja Turystyki Górskiej Zarządu Głównego PTTK	KTG ZG PTTK	Organizacja odpowiedzialna za akcję GOT PTTK ustalająca trasy oraz ich punktację.
Książeczka GOT PTTK	książeczka	Dokument uprawniający do nabycia i noszenia odznak GOT PTTK, w którym wpisuje się przebyte trasy oraz liczby punktów zdobytych do odznak GOT PTTK. Jest identyfikowana przez numer książeczki.
Odcinek		Odcinek trasy posiadający początek i koniec, za którego pokonanie przyznawane są punkty. Może być identyfikowany przez nazwę lub punkt pośredni, przez który przebiega. Znajduje się w obrębie jednego pasma górnego. Może być punktowany oficjalnie lub nie posiadać przypisanej z góry punktacji.
Odcinek oficjalny		Odcinek, który ma przypisaną oficjalną punktację. Jest pokonywany w określonym kierunku.
Odcinek trasy		Odcinek wchodzący w skład trasy wpisanej do książeczki. Jego pokonanie zajmuje określoną długość czasu.
Odcinek własny		Odcinek, który nie posiada przypisanej z góry punktacji. Punktację za odcinek własny wyznacza się na podstawie jego długości i wysokości podejścia.
Organizacja		Oficjalnie utworzona grupa ludzi współpracująca w pewnym określonym celu. Jest identyfikowana przez nazwę. RW GOT PTTK, PTTK, COTG PTTK, KTG ZG PTTK są organizacjami.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

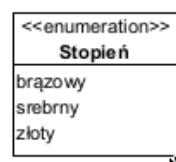
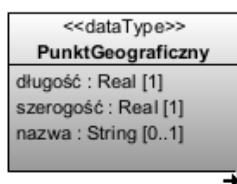
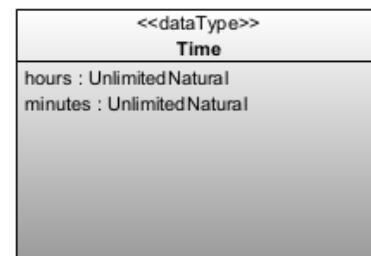
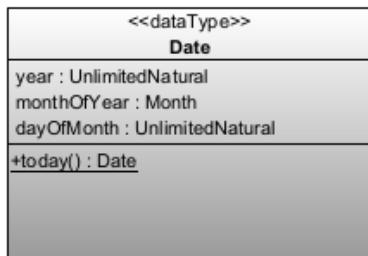
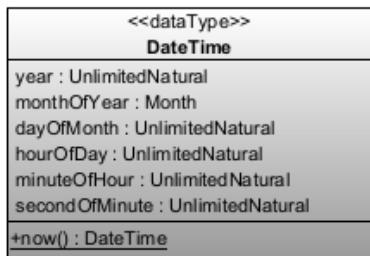
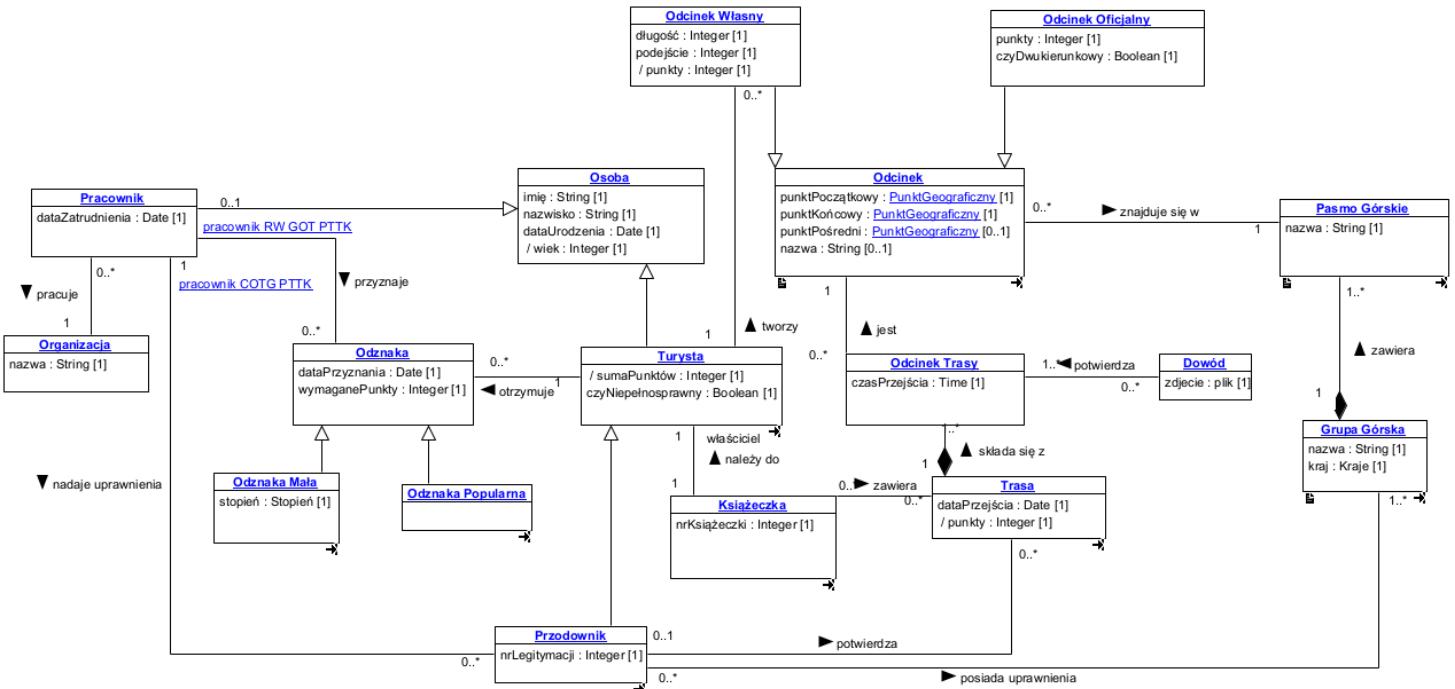
Osoba		Identyfikowana w systemie przez imię i nazwisko. Na podstawie daty urodzenia określany jest wiek osoby. Pracownicy RW GOT PTTK i COTG PTKK, członkowie KTG ZG PTTK, turyści i przodownicy turystyki górskiej są osobami.
Pasmo górskie		Grupa gór położonych jedna za drugą. Jest identyfikowane przez nazwę.
Polskie Towarzystwo Turystyczno-Krajoznawcze	PTTK	Stowarzyszenie powstałe w 1950 roku, grupujące turystów i krajoznawców. Aktualnie opiekuje się GOT PTTK.
Pracownik		Osoba zatrudniona w organizacji. Pracownicy COTG PTTK i pracownicy RW GOT PTTK są pracownikami.
Pracownik Centralnego Ośrodka Turystyki Górskiej PTTK	Pracownik COTG PTTK	Osoba zatrudniona w Centralnym Ośrodku Turystyki Górskiej PTTK, odpowiedzialna za nadawanie uprawnień przodownikom turystyki górskiej. Posiada indywidualne konto w systemie i jest do niego zalogowana. Jest pracownikiem.
Pracownik Referatu Weryfikacyjnego GOT PTTK	Pracownik RW GOT PTTK	Osoba zatrudniona w Centralnym Referacie Weryfikacyjnym lub Terenowym Referacie Weryfikacyjnym, odpowiedzialna za przyznawanie odznak turystom. Posiada indywidualne konto w systemie i jest do niego zalogowana. Jest pracownikiem.
Przodownik turystyki górskiej PTTK	Przodownik, PTG PTTK	Członek PTTK propagujący GOT PTTK, odpowiedzialny do potwierdzania tras. Może potwierdzać trasy tylko w grupie górskiej, w której ma uprawnienia. Posiada legitymację przodownicką o indywidualnym numerze. Posiada indywidualne konto w systemie i jest do niego zalogowany.
Punkt		Jednostka przyznawana za przebycie trasy na określonych zasadach, licząca się przy zdobywaniu odznaki.
Punkt geograficzny		Miejsce w terenie o określonych współrzędnych.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Rodzaj GOT	rodzaj odznaki	Szczebel w klasyfikacji Górskiej Odznaki Turystycznej.
Spis odcinków punktowanych do GOT	spis odcinków punktowanych	Lista odcinków za których przebycie została ustalona odgórna punktacja.
Stopień GOT	stopień odznaki	Szczebel w klasyfikacji Górskiej Odznaki Turystycznej, może być brązowy, srebrny lub złoty.
Szlak		Oficjalnie wytyczona droga mająca pomóc turystom orientację w terenie. Prowadzi do konkretnego punktu geograficznego, zazwyczaj będącego charakterystycznym elementem krajobrazu. Może mieć przypisaną nazwę lub kolor.
Terenowy Referat Weryfikacyjny GOT PTTK	TRW GOT PTTK	Jednostka specjalistyczna PTTK, weryfikująca książeczki GOT PTTK turystów.
Trasa		Składa się z co najmniej jednego odcinka. Jest zapisywana w książeczce GOT. Zostaje przebyta w jednym dniu. Liczba punktów za trasę jest równa sumie punktów za odcinki wchodzące w skład trasy.
Turysta		Osoba uprawiająca turystykę górską, posiadająca książeczkę GOT PTTK. Posiada indywidualne konto w systemie i jest do niego zalogowana. Może być niepełnosprawny. W danym momencie posiada pewną liczbę punktów, które zbiera w cyklu zdobywania konkretnej odznaki.
Współrzędne geograficzne	współrzędne	Szerokość i długość geograficzna wyrażone liczbą rzeczywistą określającą stopień kąta od początku układu współrzędnych geograficznych, który znajduje się w Atlantyku wschodnim. W kierunku północnym liczby dodatnie, południowym ujemne. W kierunku wschodnim liczby dodatnie, zachodnim ujemne.
Wycieczka		Potwierdzone przebycie przez turystę określonej trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Model domenowy



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Reguły biznesowe

Iden	Reguła biznesowa
<u>Reguły do generalizacji</u>	
RB01	Turysta jest osobą.
RB02	Członek KTG ZG jest osobą.
RB03	Pracownik jest osobą.
RB04	Przodownik jest turystą.
RB05	Odznaka mała jest odznaką.
RB06	Odznaka popularna jest odznaką.
RB07	Odcinek własny jest odcinkiem.
RB08	Odcinek oficjalny jest odcinkiem.
<u>Reguły do liczności asocjacji</u>	
RB09	Pracownik musi pracować w tylko jednej organizacji.
RB10	Organizacja może zatrudniać wielu pracowników.
RB11	Turysta może posiadać wiele odznak.
RB12	Odznaka może należeć do tylko jednego turysty.
RB13	Turysta może posiadać tylko jedną książeczkę.
RB14	Książeczka musi należeć do tylko jednego turysty.
RB15	W książeczkę może być wiele tras.
RB16	Trasa może być wpisana do wielu książeczek.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

RB17	Trasa musi składać się z co najmniej jednego odcinka trasy.
RB18	Odcinek trasy musi należeć do tylko jednej trasy.
RB19	Odcinek trasy może być potwierdzony co najwyżej jednym dowodem.
RB20	Dowód może potwierdzać przebycie wielu odcinków trasy.
RB21	Odcinek może wystąpić w wielu trasach.
RB22	Każdy odcinek trasy musi być jednym odcinkiem.
RB23	Odcinek musi znajdować się w tylko jednym paśmie górkim.
RB24	W jednym paśmie górkim może znajdować się wiele odcinków.
RB25	Grupa górska musi składać się z co najmniej jednego pasma górkiego.
RB26	Pasmo górskie może należeć do tylko jednej grupy górkowej.
RB27	Przodownik może potwierdzić wiele tras.
RB28	Trasa może zostać potwierdzona przez maksymalnie jednego przodownika.
RB29	Przodownik musi posiadać uprawnienia do co najmniej jednej grupy górkowej.
RB30	Do danej grupy górkowej może posiadać uprawnienia wielu przodowników.
RB31	Pracownik może przyznać wiele odznak.
RB32	Odznak może zostać przyznana przez maksymalnie jednego pracownika.
RB33	Pracownik może nadawać uprawnienia wielu przodownikom.
RB34	Przodownik musi mieć nadane uprawnienia przez tylko jednego pracownika.
RB35	Turysta może tworzyć wiele odcinków własnych.
RB36	Odcinek własny musi zostać utworzony przez tylko jednego turystę.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

<u>Ograniczenia</u>	
RB37	Turysta musi mieć co najmniej 7 lat.
RB38	Przodownik musi mieć co najmniej 18 lat.
RB39	Pracownik musi mieć co najmniej 18 lat.
RB40	Tylko pracownik zatrudniony w RW PTTK może przyznać odznaki.
RB41	Tylko pracownik zatrudniony w COTG może nadawać uprawnienia przodownikom.
RB42	Za trasę można otrzymać maksymalnie 50 punktów.
RB43	W jednym roku kalendarzowym można otrzymać wyłącznie jedną odznakę, za wyjątkiem odznaki popularnej i małej brązowej.
RB44	Punkty uzyskane za odcinki tras znajdujące się poza Polską mogą stanowić maksymalnie połowę punktów wymaganych do zdobycia odznaki.
RB45	W przypadku przebycia dwa razy tego samego odcinka w tym samym kierunku w cyklu zdobywania jednej odznaki, punkty mogą zostać przyznane tylko raz.
<u>Wnioski</u>	
RB46	Odznaka popularna musi zostać przyznana, jeżeli turysta otrzymał 60 punktów.
RB47	Odznaka mała brązowa musi zostać przyznana, jeżeli turysta otrzymał 120 punktów oraz posiada odznakę popularną.
RB48	Odznaka mała srebrna musi zostać przyznana, jeżeli turysta otrzymał 360 punktów oraz posiada odznakę małą brązową.
RB49	Odznaka mała złota musi zostać przyznana, jeżeli turysta otrzymał 720 punktów oraz posiada odznakę małą srebrną.
RB50	Jeżeli turysta jest niepełnosprawny, odznaka popularna musi zostać przyznana, jeżeli turysta otrzymał 45 punktów.
RB51	Jeżeli turysta jest niepełnosprawny, odznaka mała brązowa musi zostać przyznana, jeżeli turysta otrzymał 90 punktów oraz posiada odznakę popularną.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

RB52	Jeżeli turysta jest niepełnosprawny, odznaka mała srebrna musi zostać przyznana, jeżeli turysta otrzymał 270 punktów oraz posiada odznakę małą brązową.
RB53	Jeżeli turysta jest niepełnosprawny, odznaka mała złota musi zostać przyznana, jeżeli turysta otrzymał 540 punktów oraz posiada odznakę małą srebrną.
RB54	Trasy przebiegające przez inne niż Polska kraje mogą potwierdzać przodownicy posiadający uprawnienia na przylegające grupy górskie po stronie Polskiej.

Obliczenia

RB55	Wiek osoby musi być obliczony jako różnica lat pomiędzy datą dzisiejszą a datą urodzenia osoby.
RB56	Liczba punktów przyznana za trasę musi być obliczona jako suma punktów przyznanych za odcinki trasy, które do niej należą.
RB57	Liczba punktów za odcinek własny musi być obliczona jako suma punktów obliczanych na podstawie długości odcinka i pokowanej różnicy wysokości - punkt za każdy przebyty kilometr i punkt za każde przebyte 100m różnicy poziomów.
RB58	Liczba punktów turysty jest musi być obliczona jako suma punktów za odcinki tras, które turysta przebył.
RB59	Liczba punktów turysty musi być pomniejszana o punkty wymagane do przyznania odznaki przy jej weryfikacji. Pozostała po pomniejszeniu liczba punktów może stanowić maksymalnie połowę wymaganych punktów do zdobycia kolejnego stopnia odznaki.

Wyzwalacze

RB60	Po przekazaniu przez turystę trasę do potwierdzenia, musi ona zostać skierowana do potwierdzenia do przodownika, który posiada uprawnienia na grupy górskie, które trasa obejmuje.
RB61	Po przekazaniu przez turystę książeczkę do weryfikacji, musi ona zostać skierowana do pracownika RW PTTK w celu sprawdzenia liczby punktów i przyznania odznaki.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Wyniki etapu 2: Specyfikacja wymagań, diagram przypadków użycia, model informacyjny, prototyp interfejsu, specyfikacja przypadków użycia

E - książeczka GOT PTTK

Projektowanie oprogramowania



E - GOT PTTK

Skład zespołu:
Agnieszka Kłobus
Aleksandra Stecka
(zespół 8)

Prowadzący:
dr inż. Zbigniew Szpunar

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Zmiany do Etapu 1:

- zmieniono nazwę produktu z E - książeczka GOT PTTK na E - GOT PTTK (punkt 2.2 Etapu 1)
- dodano terminy do słownika:

Termin	Synonimy	Definicja terminu
Dane przodownika		Dowód przebycia odcinka trasy w danych identyfikujących przodownika - numeru legitymacji oraz imienia i nazwiska.
Punkt oficjalny		Miejsce w terenie o określonych współrzędnych definiowane przez Członka KTG PTTK. Posiada nazwę, która identyfikuje go w spisie wszystkich punktów oficjalnych. Między punktami oficjalnymi wyznaczane są odcinki oficjalne.
Punkt własny		Miejsce w terenie o określonych współrzędnych definiowane przez Turystę. Może wchodzić w skład odcinków własnych.
Zdjęcie		Dowód przebycia odcinka trasy w formie pliku .jpg - może przedstawiać Turystę na trasie, charakterystyczne elementy krajobrazu lub zrzut ekranu z aplikacji śledzącej lokalizację.

Specyfikacja wymagań

Potrzeby

Iden.	Opis potrzeby
P1	Gość chce przeglądać i wyszukiwać odcinki w spisie odcinków oficjalnie punktowanych.
P2	Turysta chce przeglądać i wyszukiwać odcinki w spisie odcinków oficjalnie punktowanych.
P3	Członek KTG chce przeglądać i wyszukiwać odcinki w spisie odcinków oficjalnie punktowanych.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

P4	Turysta chce posiadać książeczkę.
P5	Turysta chce przebywać oficjalnie punktowane odcinki i dostać za nie odpowiednią punktację.
P6	Turysta chce przebywać odcinki nie posiadające oficjalnej punktacji.
P7	Turysta chce, aby punktacja za własne odcinki była wyliczana automatycznie na podstawie ich długości i przebytej różnicy wysokości.
P8	Turysta chce zdobywać punkty za przebyte odcinki.
P9	Turysta chce wpisywać przebyte trasy do książeczek.
P10	Turysta chce zamieścić dowód przebycia trasy.
P11	Turysta chce otrzymać potwierdzenie przebycia trasy.
P12	Turysta chce oddać książeczkę do weryfikacji i dostać odznakę.
P13	Przodownik chce potwierdzać przebycie trasy przez turystów.
P14	Pracownik RW chce weryfikować książeczki turystów.
P15	Członek KTG chce dodawać, edytować i usuwać odcinki zapisane w spisie odcinków oficjalnie punktowanych.
P16	Członek KTG chce ustawiać status odcinków na zlikwidowane.
P17	Pracownik COTG chce nadawać uprawnienia przodownikom.
P18	Użytkownicy chcą chronić swoje dane i dostęp do operacji.
P19	Członek KTG chce tworzyć konta pracownikom COTG, pracownikom RW oraz innym członkom KTG.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Wymagania

Iden.	Tytuł	Tekst	Realizowana potrzeba
W1	Zakładanie konta	System umożliwia założenie indywidualnego konta.	P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12
W2	Przypisanie książeczki do turysty	System umożliwia przypisanie książeczki do konta turysty.	P4, P5, P6, P7, P8, P9, P10, P11, P12
W3	Logowanie	System umożliwia zalogowanie się na indywidualne konto.	P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18
W4	Wylogowanie	System umożliwia wylogowanie się z indywidualnego konta.	P18
W5	Wyświetlanie listy odcinków	System umożliwia wyświetlanie listy odcinków.	P1, P2, P3
W6	Filtrowanie odcinków	System umożliwia filtrowanie listy odcinków na podstawie zadanego kryterium.	P1, P2, P3
W7	Przebywanie tras	System umożliwia rejestrowanie przebycia trasy przez turystę.	P4, P5, P6, P9
W8	Zapisywanie trasy w książeczce	System umożliwia zapisywanie w książeczce turysty przebytych przez niego tras.	P4, P5, P6, P7, P8, P9, P10, P11
W9	Liczba punktów turysty	System sumuje punkty uzyskane za odcinki tras przebyte przez turystę i zatwierdzone przez przodownika oraz pomniejsza je o punkty wymagane do przyznania odznak, które turysta posiada.	P8, P12, P5, P7
W10	Obliczanie punktów za trasę	System sumuje punkty uzyskane za odcinki tras przebyte przez turystę.	P5, P6, P7, P8
W11	Przebywanie odcinków	System umożliwia turystie rejestrowania przebycia odcinka.	P5, P6
W12	Zarządzanie	System umożliwia turystie edytowanie, dodawanie i	P6

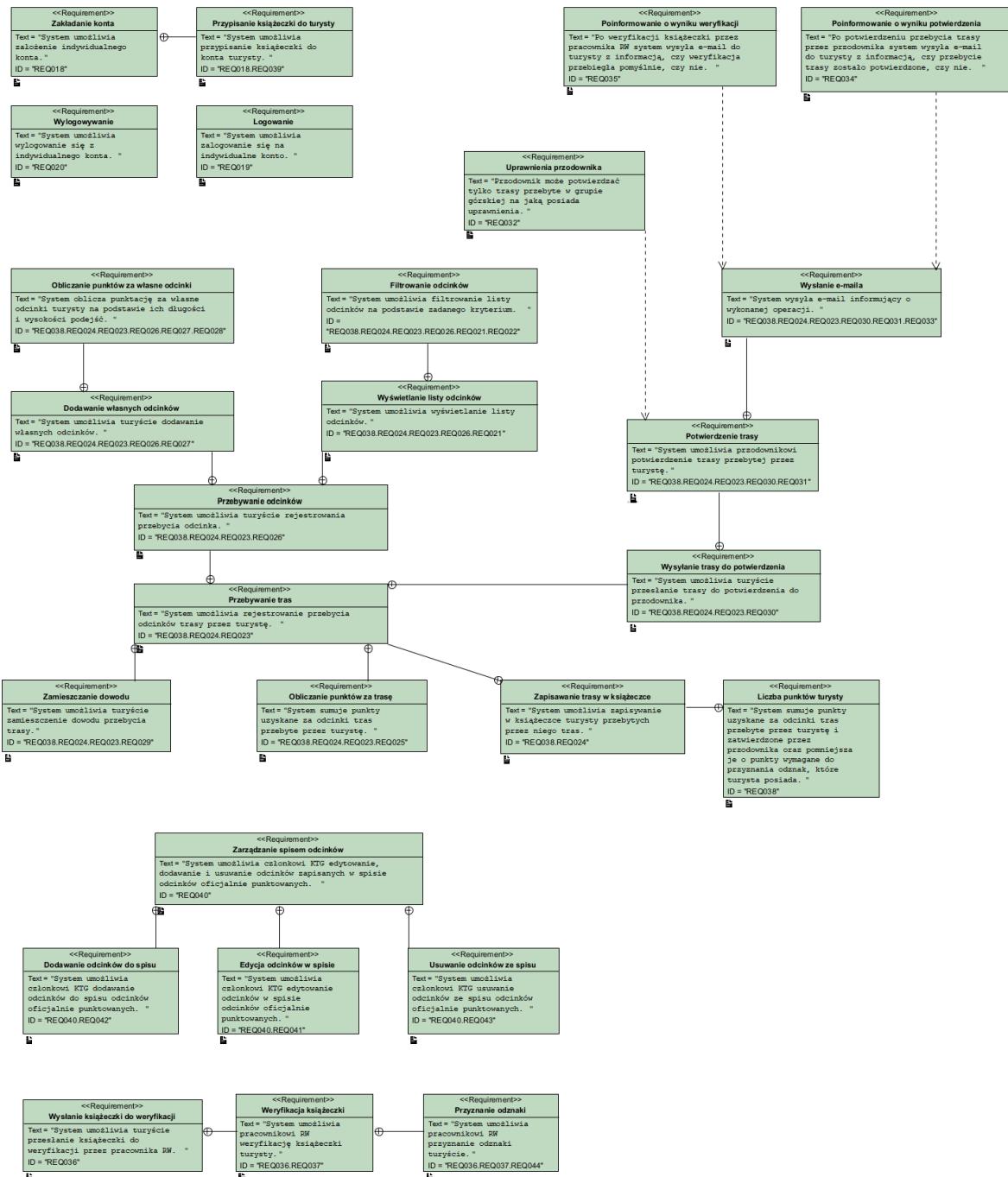
E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

	odcinkami własnymi	usuwanie odcinków zapisanych w spisie odcinków własnych.	
W13	Edycja własnych odcinków	System umożliwia turyste edytowanie odcinków w spisie odcinków własnych.	P6
W14	Dodawanie własnych odcinków	System umożliwia turyste dodawanie odcinków do spisu odcinków własnych.	P6
W15	Usuwanie odcinków ze spisu	System umożliwia turyste usuwanie odcinków ze spisu odcinków własnych.	P6
W16	Obliczanie punktów za własne odcinki	System oblicza punktację za własne odcinki turysty na podstawie ich długości i wysokości podejść.	P7
W17	Zamieszczanie dowodu	System umożliwia turyste zamieszczenie dowodu przebycia trasy.	P10
W18	Wysłanie trasy do potwierdzenia	System umożliwia turyste przesłanie trasy do potwierdzenia do przodownika.	P11
W19	Potwierdzenie trasy	System umożliwia przodownikowi potwierdzenie trasy przebytej przez turystę.	P11, P13
W20	Uprawnienia przodownika	Przodownik może potwierdzać tylko trasy przebyte w grupie górskiej na jaką posiada uprawnienia.	P13
W21	Wysłanie książeczki do weryfikacji	System umożliwia turyste przesłanie książeczki do weryfikacji przez pracownika RW.	P12
W22	Weryfikacja książeczki	System umożliwia pracownikowi RW weryfikację książeczki turysty.	P12, P14
W23	Przyznanie odznaki	System umożliwia pracownikowi RW przyznanie odznaki turystie.	P12, P14
W24	Wysłanie e-maila	System wysyła e-mail informujący o wykonanej operacji.	P11, P12
W25	Poinformowanie o wyniku potwierdzenia	Po potwierdzeniu przebycia trasy przez przodownika system wysyła e-mail do turysty z informacją, czy przebycie trasy zostało potwierdzone, czy nie.	P11

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

W26	Poinformowanie o wyniku weryfikacji	Po weryfikacji książeczki przez pracownika RW system wysyła e-mail do turysty z informacją, czy weryfikacja przebiegła pomyślnie, czy nie.	P12
W27	Zarządzanie spisem odcinków	System umożliwia członkowi KTG edytowanie, dodawanie i usuwanie odcinków zapisanych w spisie odcinków oficjalnie punktowanych.	P15, P16, P19
W28	Edycja odcinków w spisie	System umożliwia członkowi KTG edytowanie odcinków w spisie odcinków oficjalnie punktowanych.	P15, P16, P19
W29	Dodawanie odcinków do spisu	System umożliwia członkowi KTG dodawanie odcinków do spisu odcinków oficjalnie punktowanych.	P15, P19
W30	Usuwanie odcinków ze spisu	System umożliwia członkowi KTG usuwanie odcinków ze spisu odcinków oficjalnie punktowanych.	P15, P19
W31	Nadanie uprawnień przodownikom	System umożliwia pracownikowi COTG nadanie uprawnień przodownika wybranemu turyście.	P17, P19
W32	Tworzenie kont użytkowników	System umożliwia członkowi KTG utworzenie kont pracownikom COTG, pracownikom RW oraz innym członkom KTG.	P13, P14, P15, P16, P17, P18, P19

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Model przypadków użycia i prototypy interfejsów

1. Diagram przypadków użycia

Iden	Aktorzy	Nazwa PU	Opis PU	Realizowane wymaganie
PU1	Gość	Zakładanie konta	Gość zakłada indywidualne konto w systemie. Po zalogowaniu na utworzone konto będzie traktowany jako Turysta.	W1
PU2	Gość	Logowanie	Gość loguje się na istniejące w systemie konto za pomocą swojego loginu i hasła. Po zalogowaniu zaczyna być traktowany jako Turysta, Przodownik, Pracownik lub Członek KTG ZG PTTK.	W3
PU3	Użytkownik	Wylogowanie	Użytkownik wylogowuje się ze swojego indywidualnego konta i zaczyna być ponownie traktowany jako gość.	W4
PU4	Gość lub Użytkownik	Wyświetlanie oficjalnego spisu odcinków	Gość lub Użytkownik przeglądają listę oficjalnie punktowanych odcinków wraz z danymi dotyczącymi każdego odcinka (punkt początkowy i końcowy, długość, wysokość podejścia, przypisane punkty)	W5
PU5	Gość lub Użytkownik	Filtrowanie odcinków	Gość lub Użytkownik zadają kryterium, na podstawie którego filtrowana jest lista odcinków. Wynikiem filtrowania jest lista odcinków spełniających zadane kryterium.	W6

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

PU6	Turysta	Wyświetlanie przebytych tras	Turysta przegląda przebyte trasy, które uprzednio zpisał w swojej książeczce.	W2, W7, W8, W11
PU7	Turysta	Zapisanie trasy w książeczce	Turysta tworzy i zapisuje przebytą trasę w swojej książeczce.	W2, W7, W8, W11
PU8	Turysta	Wyświetlanie szczegółów trasy	Turysta przegląda szczegóły trasy – datę przebycia trasy, liczbę punktów za trasę, listę odcinków należących do trasy, informację, czy trasa została przekazana do potwierdzenia, status trasy (potwierdzona, niepotwierdzona) – oraz zyskuje dostęp do zarządzania przebytą trasą.	W2, W7, W8, W11
PU9	Turysta	Edycja trasy	Turysta edytuje uprzednio zisaną trasę. Turysta może edytować uprzednio zisaną trasę tylko jeżeli nie została ona przekazana do potwierdzenia.	W2, W7, W8, W11
PU10	Turysta	Usunięcie trasy	Turysta usuwa uprzednio zisaną trasę. Turysta może usunąć uprzednio zisaną trasę tylko jeżeli nie została ona przekazana do potwierdzenia.	W2, W7, W8, W11
PU11	Turysta	Zamieszczenie dowodu	Turysta dodaje dowód potwierdzający przebycie trasy. Dowód ma postać obrazu lub imienia, nazwiska i numeru legitymacji Przodownika, który towarzyszył turyście w przebywaniu trasy. Turysta może dodać nowy dowód, zobaczyć dowód uprzednio dodany do trasy lub usunąć dowód uprzednio dodany do trasy.	W2, W17

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

PU12	Turysta	Wysłanie trasy do potwierdzenia	Turysta prosi Przodownika o potwierdzenie przebycia trasy.	W2, W18
PU13	Turysta	Wyświetlanie listy odcinków własnych	Turysta przegląda odcinki własne, które wcześniej utworzył w swoim koncie.	W2, W12
PU14	Turysta	Zarządzanie odcinkami własnymi	Turysta zarządza spisem odcinków własnych – może dodawać nowe odcinki, a także edytować i usuwać uprzednio dodane odcinki.	W2, W12
PU15	Turysta	Dodanie własnego odcinka	Turysta dodaje własny odcinek. Podaje punkt początkowy i końcowy danego odcinka oraz długość i wysokość podejścia.	W2, W12, W14
PU16	Turysta	Edycja własnego odcinka	Turysta edytuje uprzednio dodany własny odcinek. Turysta może edytować własny odcinek tylko, jeśli nie został on dodany do żadnej trasy.	W2, W12, W13
PU17	Turysta	Usunięcie własnego odcinka	Turysta edytuje uprzednio dodany własny odcinek. Turysta może usunąć własny odcinek tylko, jeśli nie został on dodany do żadnej trasy.	W2, W12, W15

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

PU18	Turysta	Wysłanie książeczki do weryfikacji	Turysta prosi Pracownika RW PTTK o weryfikację wypełnionej elektronicznej książeczki i przyznanie odznaki.	W2, W18
PU19	Przodownik	Wyświetlanie tras do potwierdzenia	Przodownik przegląda listę tras, które Turyści przekazali mu do potwierdzenia. Przodownik może przeglądać jedynie trasy, które znajdują się w grupie górskiej, do której Przodownik ma uprawnienia.	W19, W20
PU20	Przodownik	Wyświetlenie szczegółów trasy	Przodownik wyświetla szczegóły trasy, przekazanej mu do potwierdzenia przez Turystę. W szczegółach trasy znajdują się lista odcinków trasy, data przebycia trasy i dowody przebycia trasy.	W19, W20
PU21	Przodownik	Potwierdzenie trasy	Przodownik potwierdza trasę, przekazaną mu do potwierdzenia przez Turystę. Wynik potwierdzenia trasy może być pozytywny – Przodownik potwierdził przebycie trasy przez Turystę – lub negatywny – Przodownik zdecydował, że Turysta nie przebył trasy.	W19, W20
PU22	Pracownik RW	Wyświetlenie listy książeczek do weryfikacji	Pracownik RW przegląda listę książeczek, które Turyści przekazali mu do weryfikacji.	W22, W23
PU23	Pracownik RW	Wyświetlenie szczegółów książeczki	Pracownik RW wyświetla szczegóły książeczki, przekazanej mu do weryfikacji przez Turystę. W szczegółach książeczki znajdują się imię i nazwisko Turysty, liczba punktów Turysty, zdobyte przez niego odznaki i przebyte przez niego trasy.	W22, W23

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

PU24	Pracownik RW	Weryfikacja książeczkki	Pracownik RW wyświetla szczegóły książeczkki Turysty w celu potwierdzenia spełnienia warunków otrzymania odznaki. Weryfikacja może mieć wynik pozytywny, czyli przyznanie odznaki, lub negatywny, czyli odrzucenie wniosku.	W22, W23
PU25	Pracownik RW	Przyznanie odznaki	Pracownik RW przyznaje odznakę Turyście po pomyślnej weryfikacji książeczkki.	W23
PU26	Pracownik COTG	Wyszukanie Turysty	Pracownik COTG wyszukuje Turystę, dzięki czemu może nadać dodatkowe uprawnienia konkretnemu Turyście.	W20, W31
PU27	Pracownik COTG	Nadanie uprawnień Przodownikowi	Pracownik COTG nadaje Turyście uprawnienia Przodownika, co umożliwia mu wykonywanie dodatkowych akcji w systemie.	W31
PU28	Pracownik COTG	Nadanie uprawnień w grupie górskiej	Pracownik COTG nadaje Przodownikowi uprawnienia w grupie górskiej, dzięki czemu Przodownik może potwierdzać przebycie tras znajdujących się w tej grupie górskiej.	W20
PU29	Członek KTG PTTK	Zarządzanie oficjalnym spisem odcinków	Członek KTG PTTK zarządza oficjalnym spisem odcinków – może dodawać nowe odcinki, a także edytować i usuwać uprzednio dodane odcinki.	W27

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

PU30	Członek KTG PTTK	Edycja odcinka w spisie	Członek KTG edytuje odcinek w spisie - zmienić jego status na zlikwidowany, zmienić jego punktację itd.	W27, W28
PU31	Członek KTG PTTK	Dodanie odcinka do spisu	Członek KTG może dodać nowy odcinek do listy, ustawiając jego punkty początkowy, końcowy i pośredni oraz wyznaczając przydzieloną za niego liczbę punktów.	W27, W29
PU32	Członek KTG PTTK	Usunięcie odcinka ze spisu	Członek KTG usuwa odcinek ze spisu jeśli żaden Turysta nie wpisał trasy z tym odcinkiem do swojej książeczki.	W27, W30
PU33	Członek KTG PTTK	Tworzenie konta użytkownika	Członek KTG tworzy konto dla Pracownika RW, Pracownika COTG lub innego Członka KTG. Drugi użytkownik podaje hasło i login do swojego konta.	W32
PU34	System	Obliczenie punktów turysty	System oblicza liczbę aktualnych punktów na koncie Turysty.	W9
PU35	System	Obliczenie punktów za trasę	System oblicza liczbę punktów, które przysługują Turystie za przebycie wszystkich odcinków, wchodzących w skład trasy.	W10

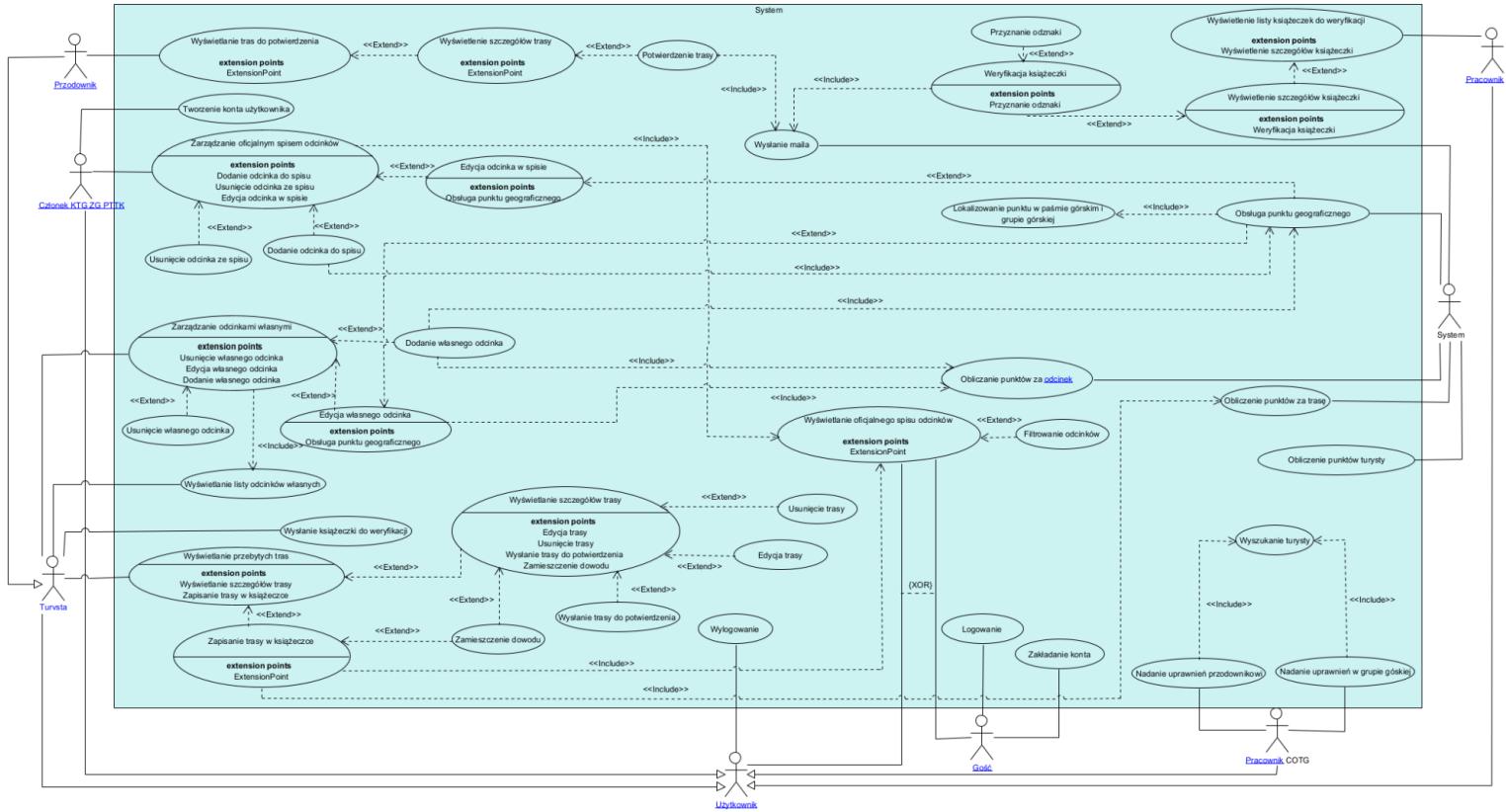
E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

PU36	System	Obliczenie punktów za odcinek	System oblicza liczbę punktów, które przysługują Turystie za przebycie jego odcinka o określonej długości i wysokości podejścia.	W11, W13, W14
PU37	System	Wysłanie e-maila	System wysyła e-mail do Turysty, aby poinformować go o wyniku potwierdzenia trasy lub weryfikacji książeczki, o które poprosił Turysta.	W24, W25, W26
PU38	System	Obsługa punktu geograficznego	System przechowuje spis punktów wykorzystywanych przez Turystów w swoich odcinkach własnych oraz przez Członków KTG PTTK w odcinkach oficjalnie punktowanych.	W12, W13, W14, W27, W28, W29
PU39	System	Lokalizowanie punktu w paśmie górkim	System lokalizuje punkt w paśmie górkim na podstawie jego współrzędnych oraz w grupie górskiej na podstawie znalezionej pasma górkiego.	W12, W13, W14, W27, W28, W29

E-książeczka GOT PTTK

Projekt

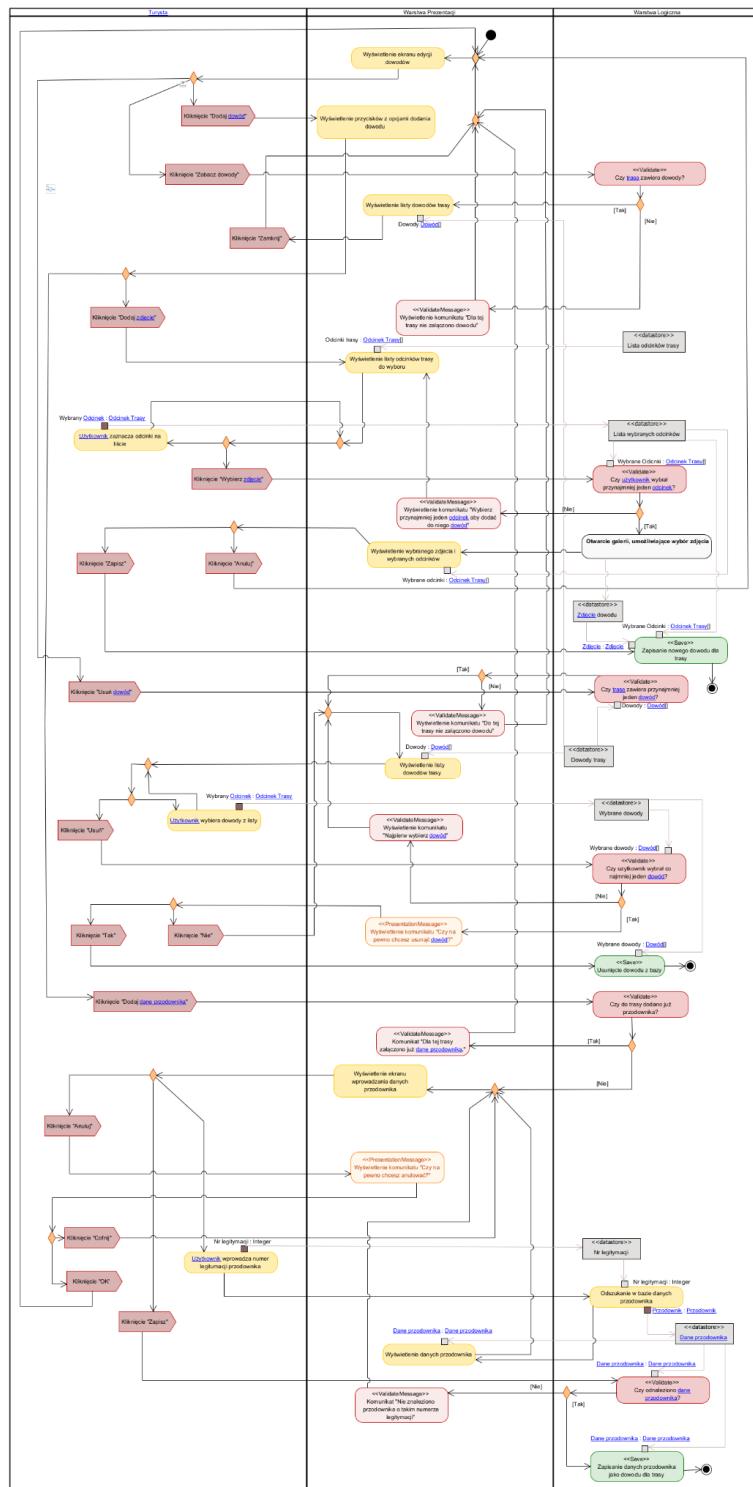
Data: 18.01.2021 r.



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

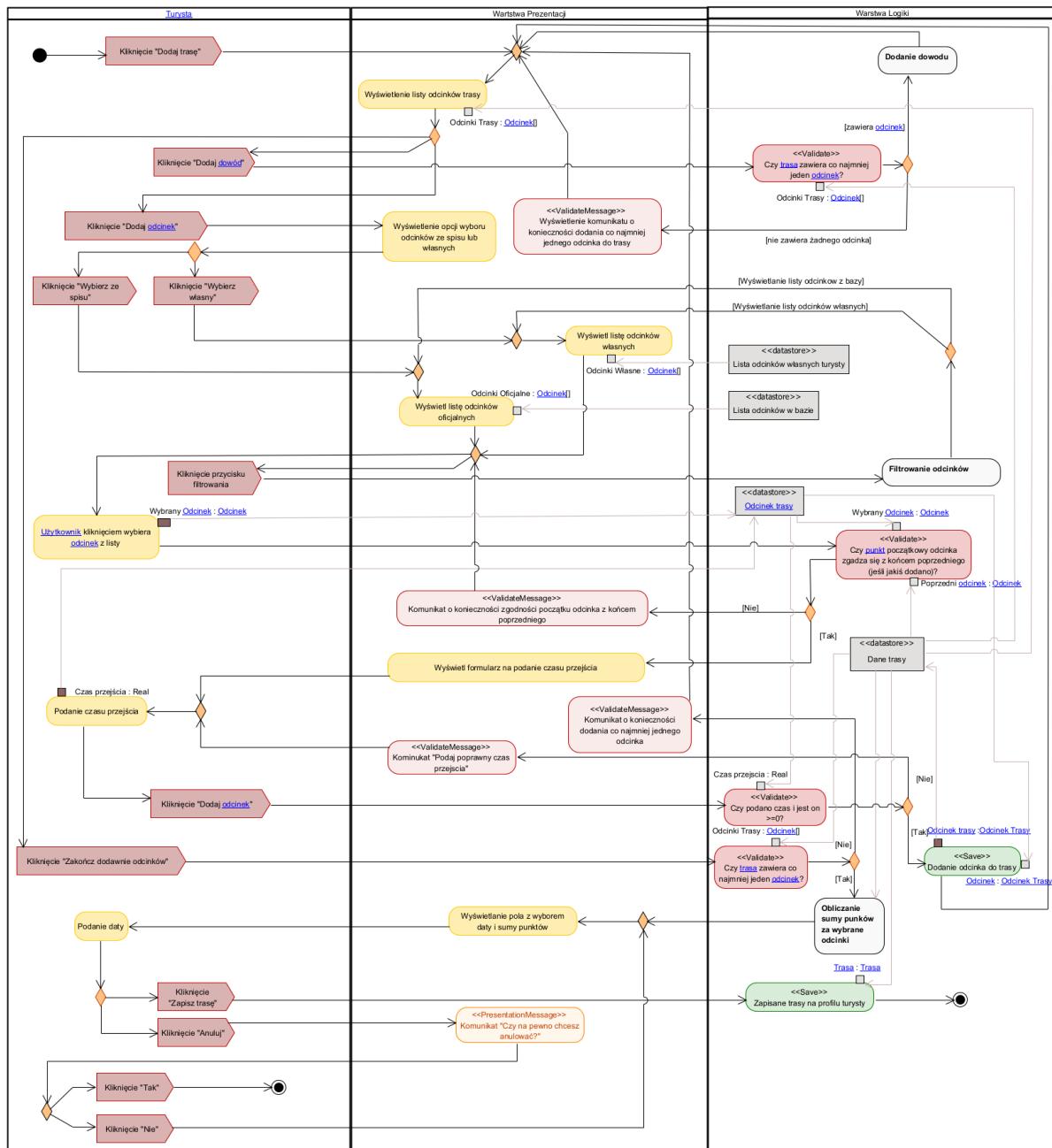
2. Specyfikacja przypadków użycia - Agnieszka Kłobus

2.1. Zamieszczenie dowodu



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

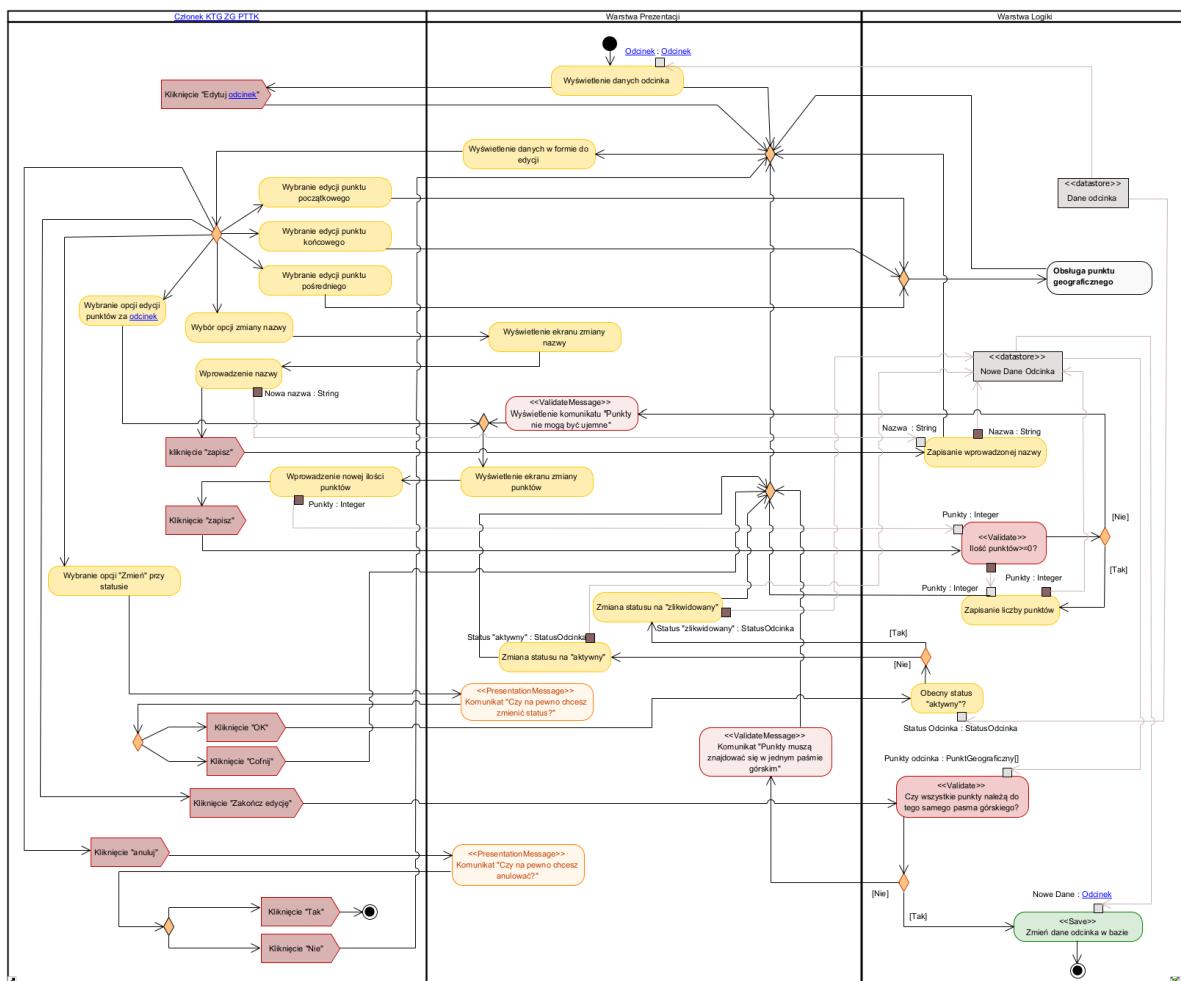
2.2. Zapisanie trasy w książeczkę



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

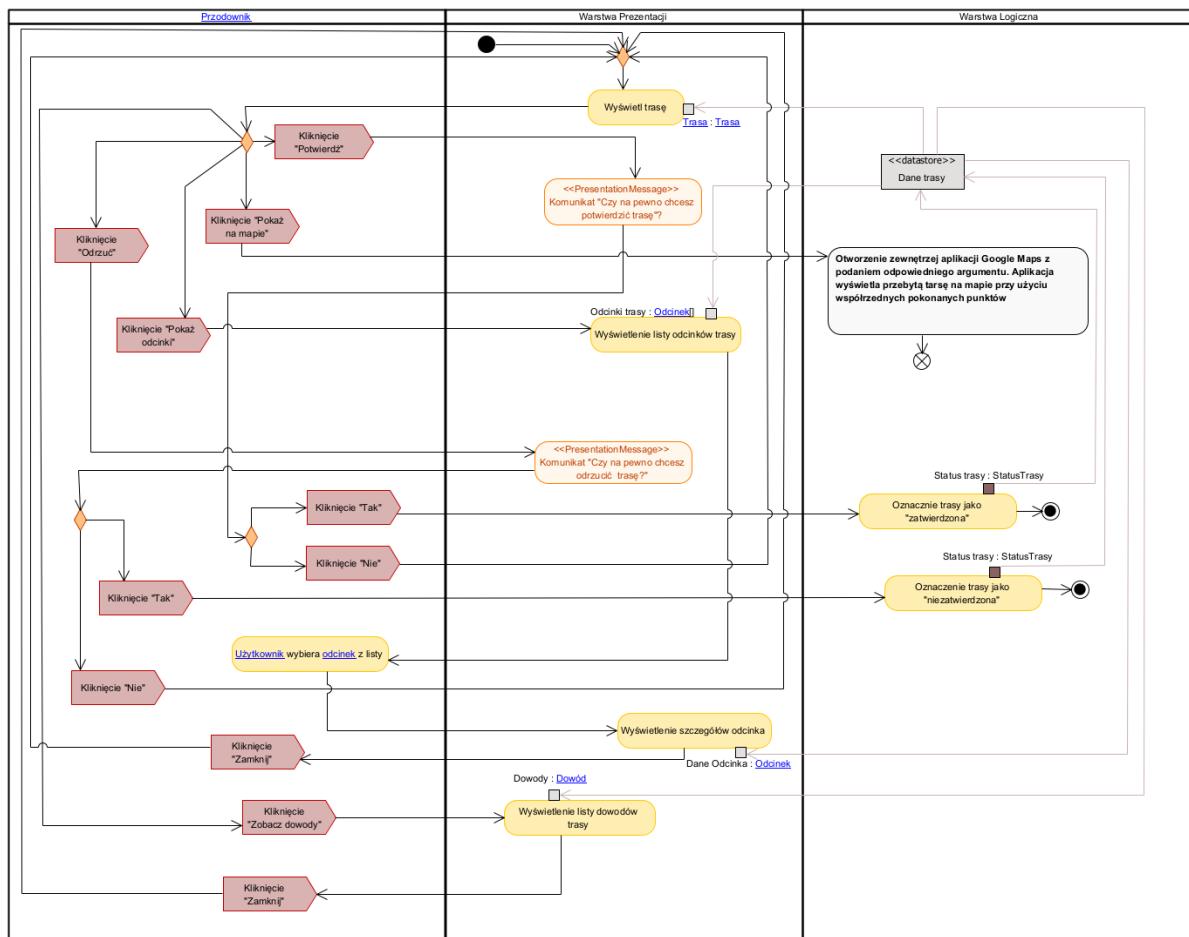
2. Specyfikacja przypadków użycia - Aleksandra Stecka

2.1. Edycja odcinka w spisie



E-ksiązeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

2.2. Potwierdzenie trasy

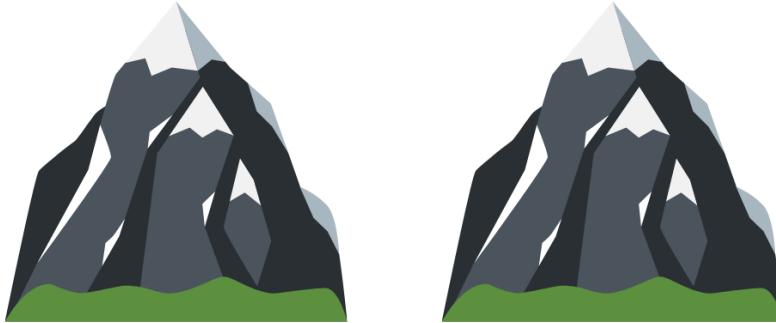


E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

3. Prototyp interfejsu

3.1. Reguły przewodnika stylu

3.1.1. Logo w wersji z podpisem i bez podpisu:



E - GOT PTTK

3.1.2. W lewym rogu górnego paska ekranu zawsze nazwa produktu.

3.1.3. Wykorzystywane kolory:

- 3.1.3.1. Biały
- 3.1.3.2. Czarny
- 3.1.3.3. Szary
- 3.1.3.4. Zielony
- 3.1.3.5. Czerwony
- 3.1.3.6. Cyan
- 3.1.3.7. Pomarańczowy
- 3.1.3.8. Fioletowy
- 3.1.3.9. Różowy
- 3.1.3.10. Żółty

3.1.4. Wizualne oddzielenie użytkowników:

- 3.1.4.1. Turysta – Cyan
- 3.1.4.2. Członek KTG ZG PTTK – Pomarańczowy
- 3.1.4.3. Pracownik COTG – Fioletowy
- 3.1.4.4. Przodownik – Różowy
- 3.1.4.5. Gość – Żółty

3.1.5. Wykorzystywana czcionka:

3.1.5.1. normalna:

ABCDEFGHIJKLMNOPRSTUVXYZĄĆŁÓŚŻŻ
abcdefghijklmnoprstuvwxyzząćłóśżż
1234567890!@#\$%^&*()

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

3.1.5.2. pogrubiona:

ABCDEFGHIJKLMNOPRSTUVXYZĄĘŁÓŞŻŻ
abcdefghijklmnoprstuvwxyząęłóşżż
1234567890!@#\$%^&*()

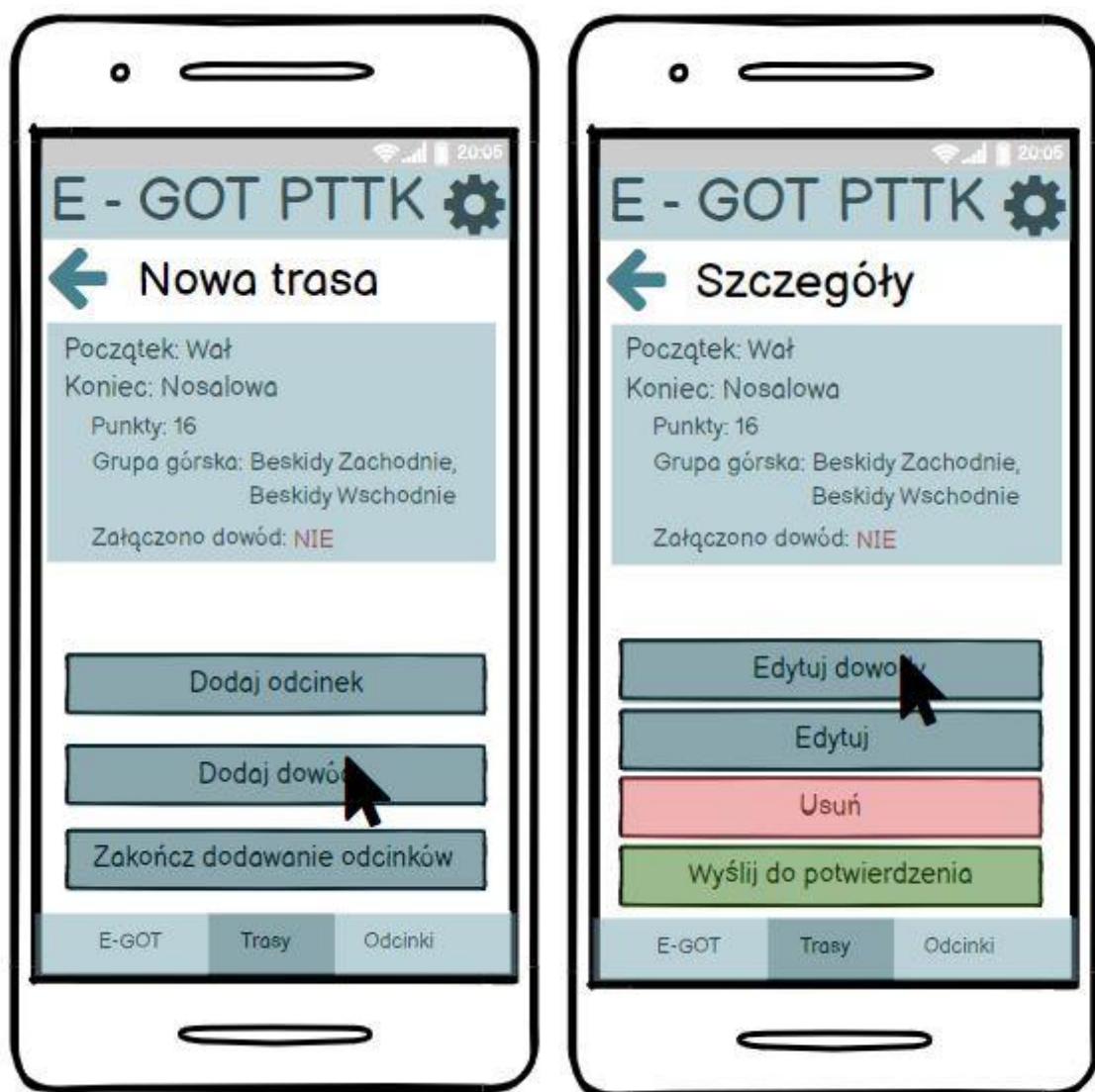
3.1.5.3. Na prototypach wykorzystywana czcionka dostępna w narzędziu Balsamiq.

- 3.1.6. Przycisk Zapisz zawsze w kolorze zielonym, Anuluj zawsze w kolorze czerwonym.
- 3.1.7. Przycisk OK zawsze w kolorze zielonym, Cofnij zawsze w kolorze czerwonym.
- 3.1.8. Słowa kluczowe TAK, AKTYWNY, ZATWIERDZONA zawsze w kolorze zielonym, NIE, ZLIKWIDOWANY, NIEZATWIERDZONA zawsze w kolorze czerwonym.
- 3.1.9. Na dole każdego ekranu pasek nawigacyjny z dostępnymi zakładkami, w kolorze odpowiedniego użytkownika.
- 3.1.10. Tło ekranów zawsze w kolorze białym.
- 3.1.11. Przyciski zawsze w kolorze odpowiedniego użytkownika.
- 3.1.12. Czcionka danych wprowadzanych przez użytkownika w kolorze odpowiedniego użytkownika.
- 3.1.13. W prawym rogu górnego paska ekranu zawsze przycisk przekierowujący do Ustawień.
- 3.1.14. Tytuł sekcji zawsze w lewym górnym rogu ekranu.
- 3.1.15. Powrót do poprzedniego ekranu zawsze przed tytułem sekcji w lewym górnym rogu ekranu.
- 3.1.16. Powrót do poprzedniego ekranu zawsze w kolorze odpowiedniego użytkownika.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

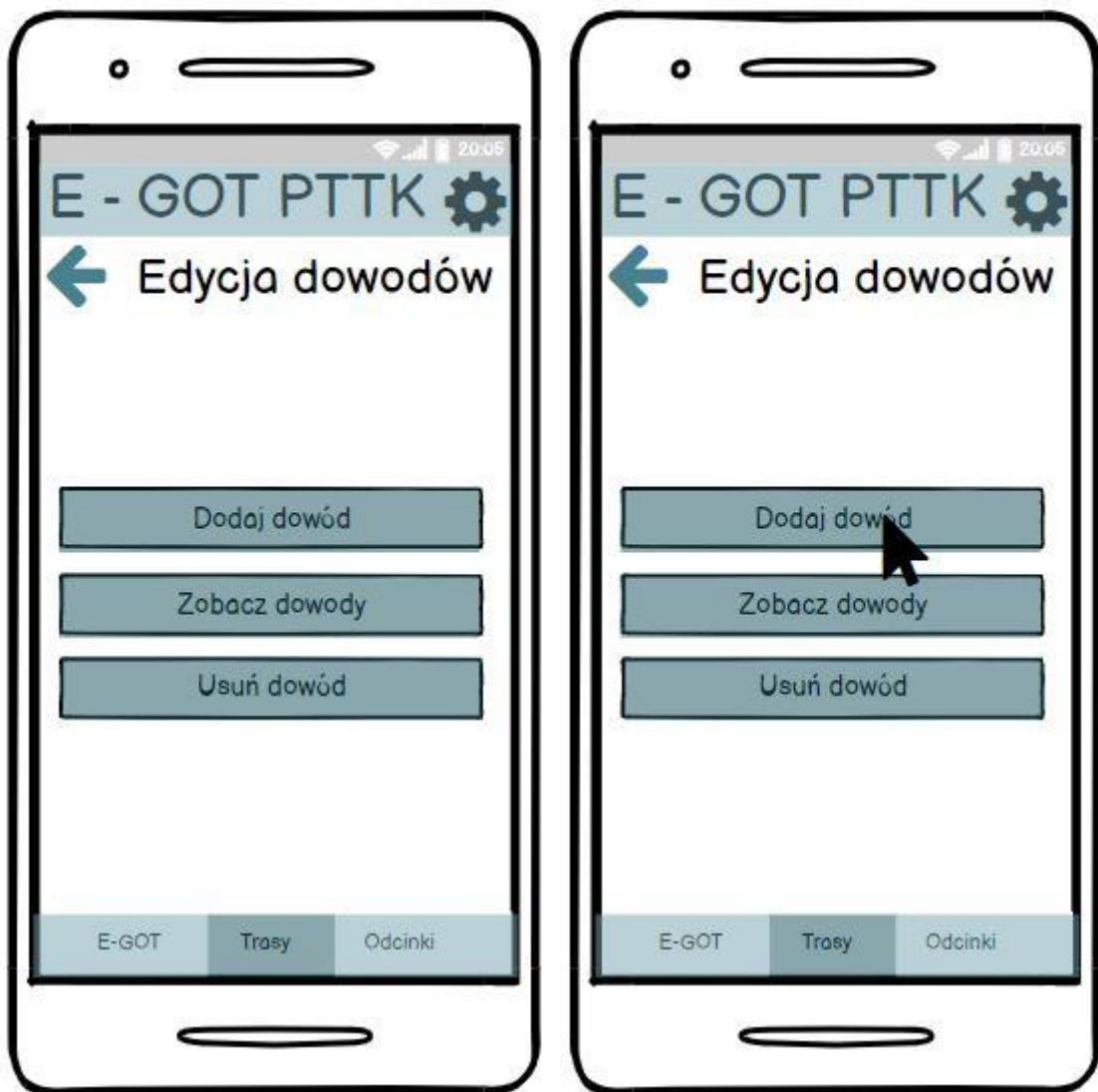
3. Prototyp interfejsu - Agnieszka Kłobus

3.1. Zamieszczenie dowodu



Rys 1. Uruchamianie PU na dwa sposoby – dowód można dodać od razu dodając trasę lub później wyświetlać szczegółы tras.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



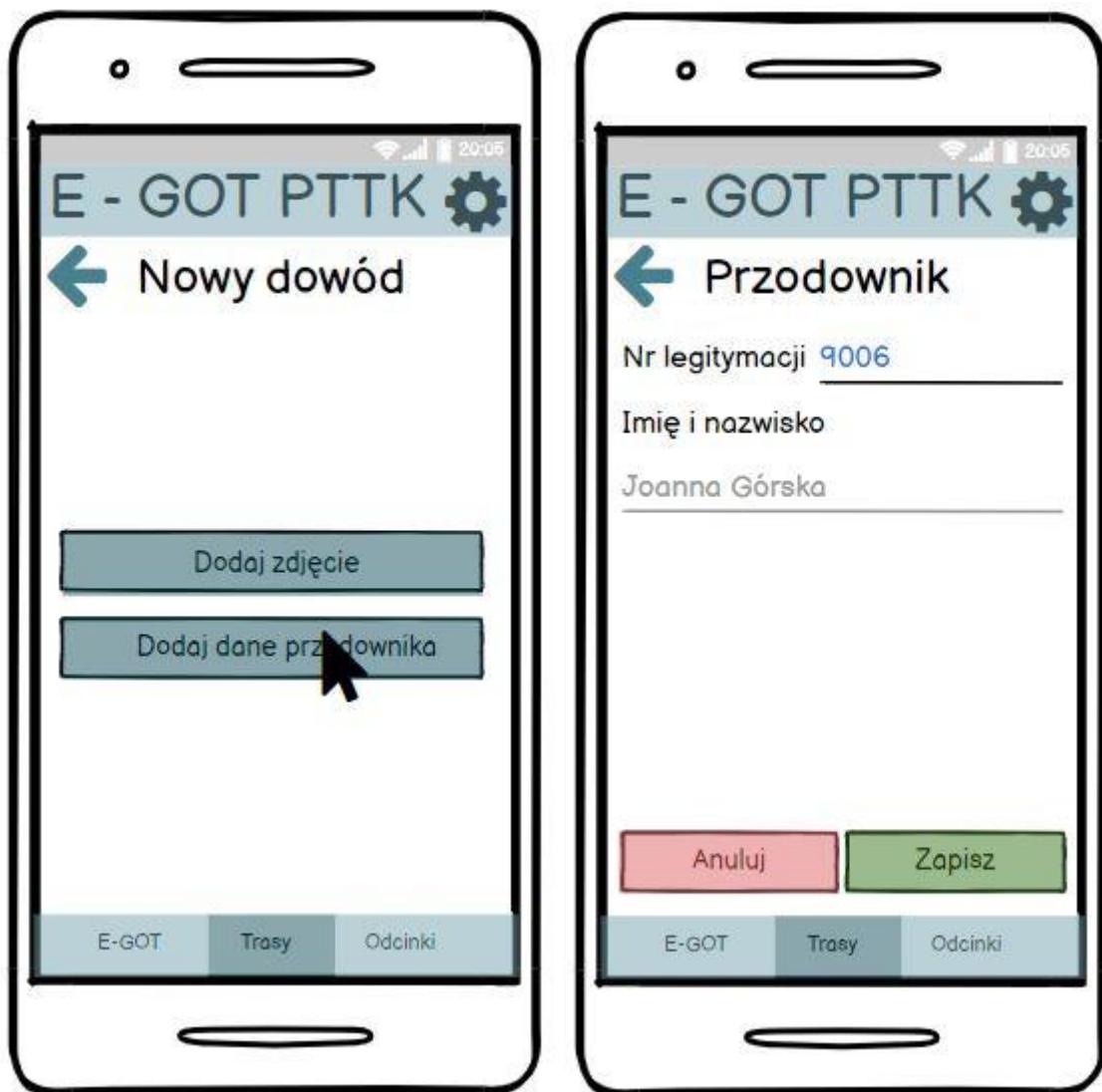
Rys 2. Scenariusz główny – widok główny zamieszczania dowodu. W ramach PU zamieszczanie dowodu można dodać dowód, zobaczyć dowody uprzednio dodane do trasy lub usunąć dowody uprzednio dodane do trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 3. Scenariusz główny – po kliknięciu przycisku **Dodaj dowód** pojawia się możliwość wyboru między dodaniem dowodu w formie zdjęcia lub w formie danych przodownika, który uczestniczył w wycieczce.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 4. Scenariusz alternatywny – po kliknięciu przycisku **Dodaj dane przodownika** pojawia się ekran umożliwiający podanie numeru legitymacji przodownika. System uzupełnia imię i nazwisko przodownika na podstawie podanego numeru legitymacji.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 5. Scenariusz alternatywny – po uzupełnieniu danych przodownika i kliknięciu przycisku **Zapisz** następuje zapisanie dowodu i powrót do ekranu głównego zamieszczania dowodu.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



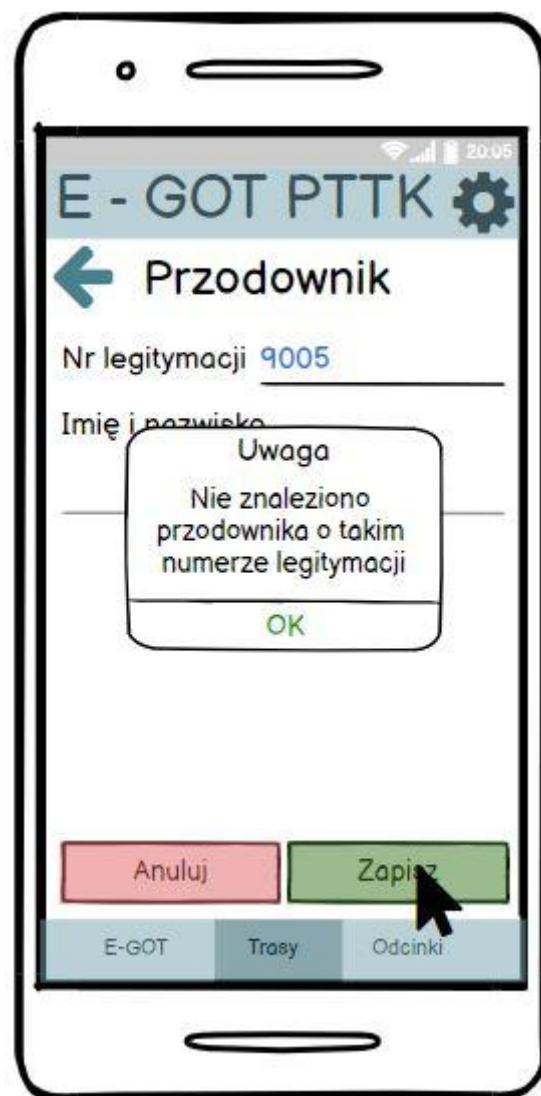
Rys 6. Wyjątek – próba dodania dowodu w formie danych przodownika dla trasy, dla której załączono już dowód w formie danych przodownika.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



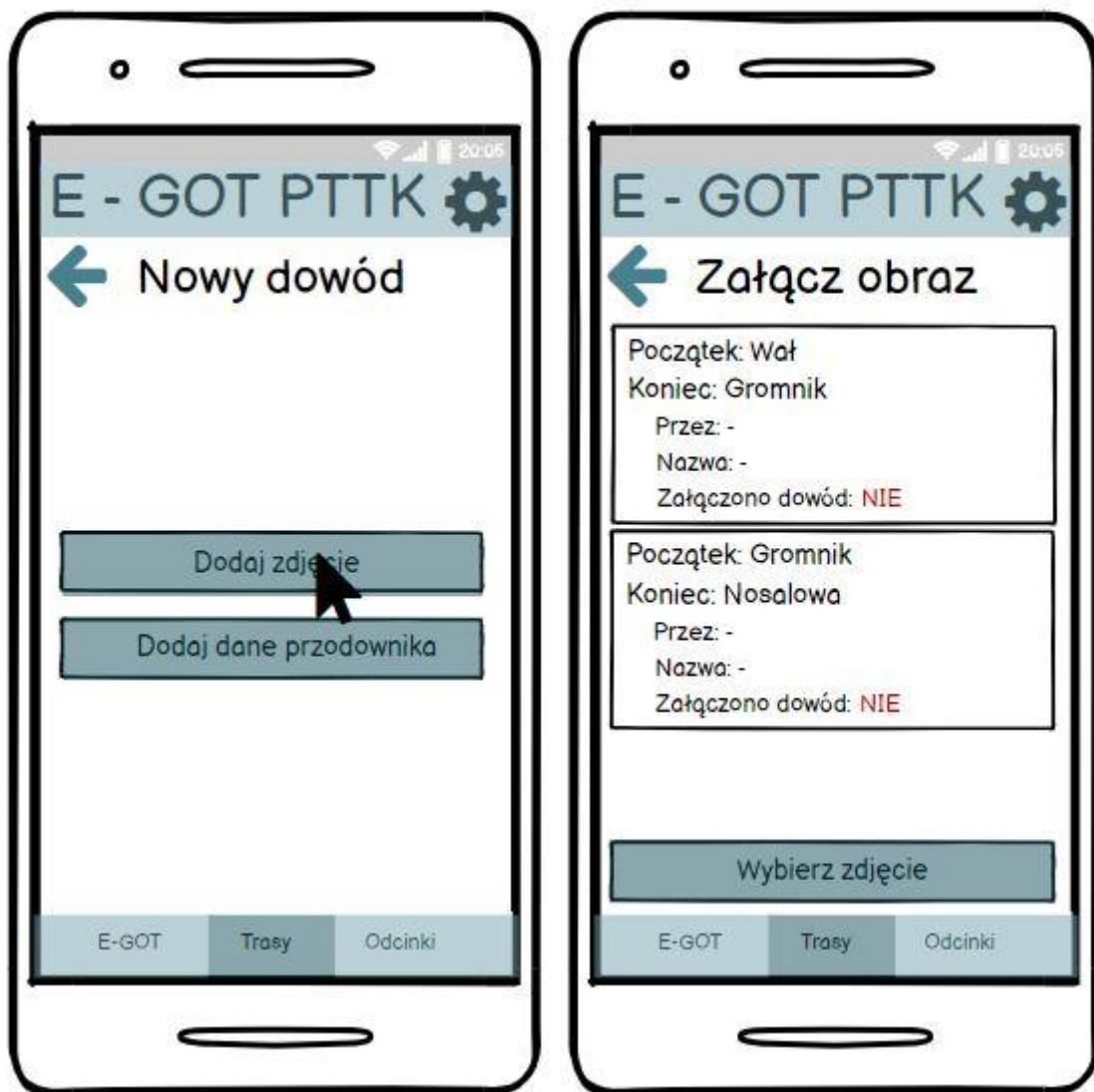
Rys 7. Wyjątek – naciśnięcie przycisku **Anuluj**. Po naciśnięciu przycisku **Anuluj** pojawia się komunikat potwierdzenia chęci anulowania. Po naciśnięciu przycisku **OK** dodawanie nowego dowodu jest anulowane, a po naciśnięciu przycisku **Cofnij** dodawanie nowego dowodu nie jest anulowane i chowa się komunikat.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



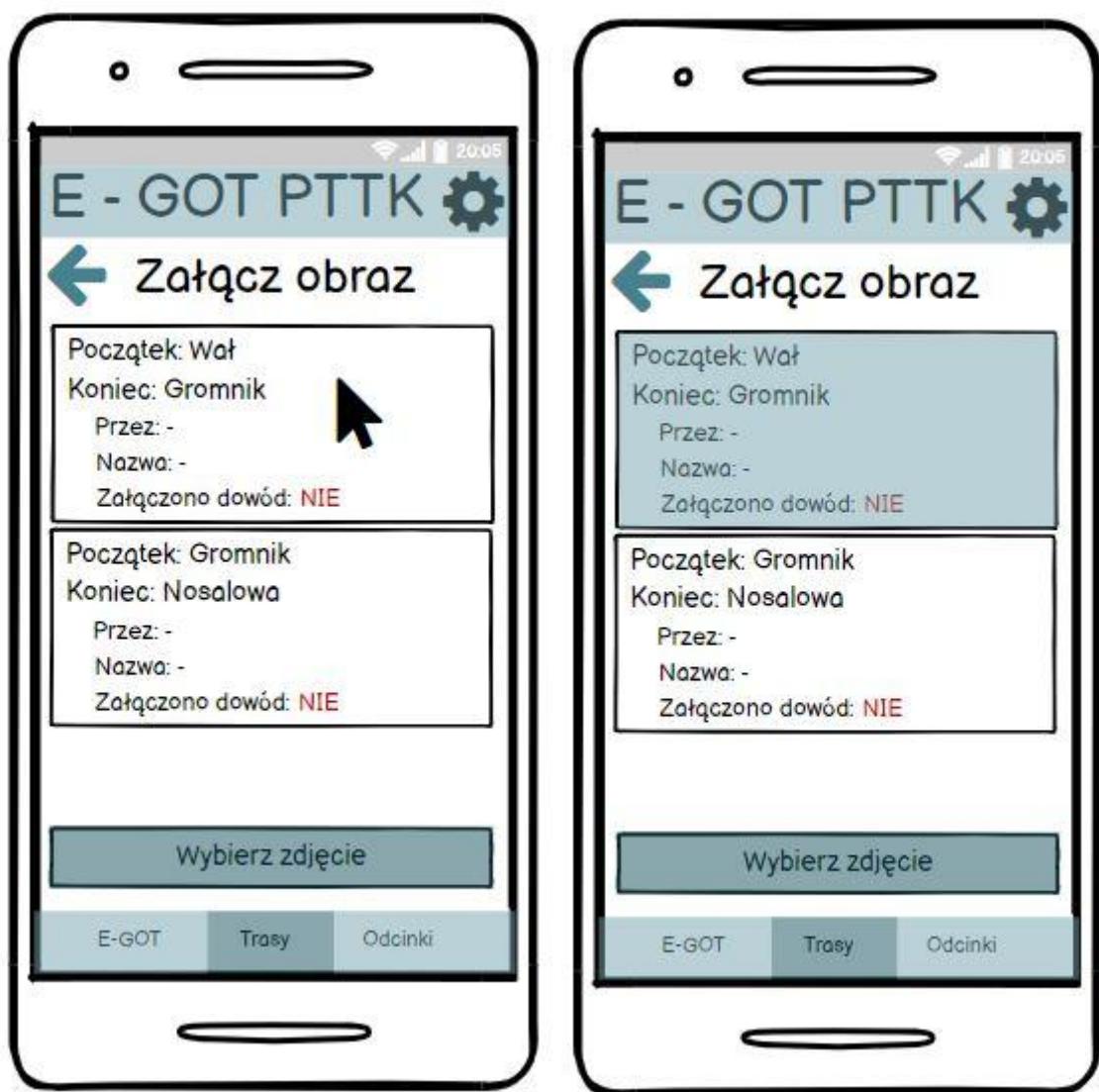
Rys 8. Wyjątek – próba podania numeru legitymacji przodownika, który nie istnieje.
Aplikacja nie umożliwia zapisania danych przodownika w takiej sytuacji.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



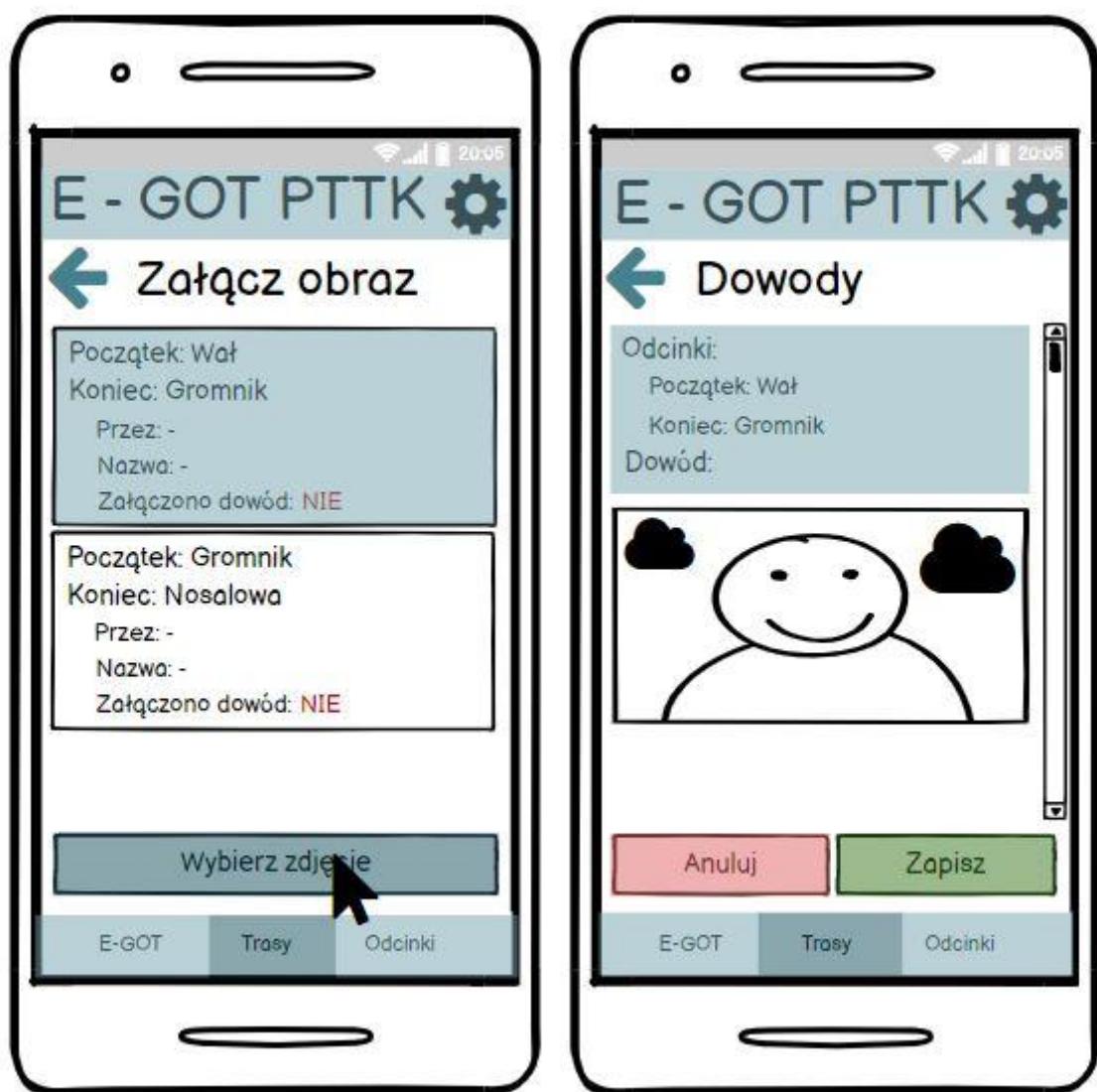
Rys 9. Scenariusz główny – po kliknięciu przycisku **Dodaj zdjęcie** pojawia się ekran główny dodawania dowodu w formie zdjęcia. Wyświetlana jest lista wszystkich odcinków trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 10. Scenariusz główny – aby załączyć dowód w formie zdjęcia należy wybrać odcinek lub odcinki, których przebycie ma dowodzić załączane zdjęcie.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 11. Scenariusz główny – po wybraniu odcinka i kliknięciu przycisku **Wybierz zdjęcie** uruchamiana jest aplikacja do przeglądania zdjęć (na przykład Galeria), która umożliwia wybór zdjęć. Po wybraniu zdjęć wyświetlana jest lista dowodów.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



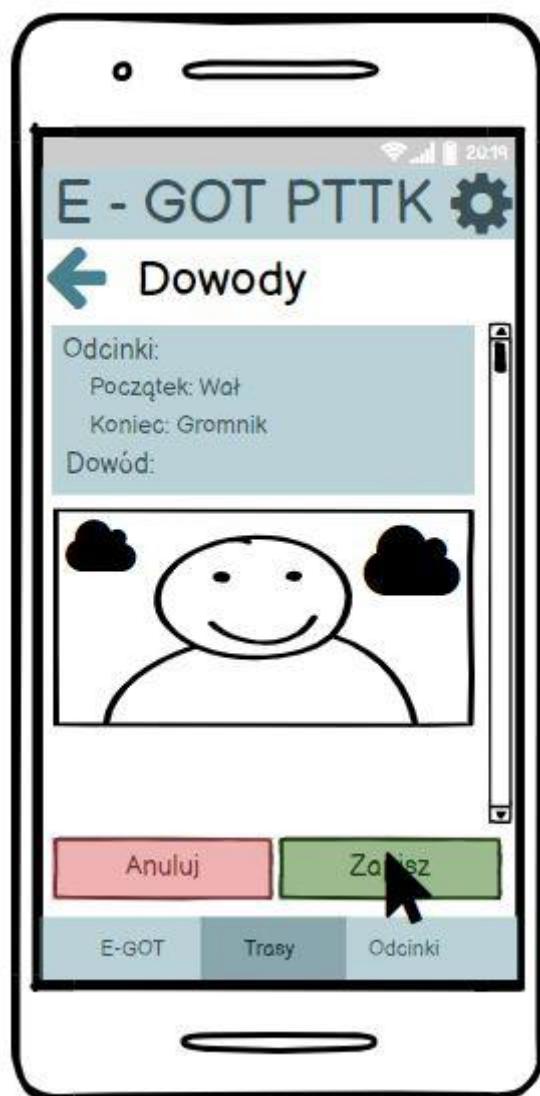
Rys 12. Wyjątek – próba załączania zdjęcia bez wybrania odcinka. Aplikacja nie umożliwia wyboru zdjęcia, jeśli nie wiadomo, którego odcinka przebycie ma potwierdzać.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 13. Wyjątek – naciśnięcie przycisku **Anuluj**. Po naciśnięciu przycisku **Anuluj** pojawia się komunikat potwierdzenia chęci anulowania. Po naciśnięciu przycisku **OK** dodawanie nowego dowodu jest anulowane, a po naciśnięciu przycisku **Cofnij** dodawanie nowego dowodu nie jest anulowane i chowa się komunikat.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 14. Scenariusz główny – po uzupełnieniu wyborze zdjęcia i kliknięciu przycisku **Zapisz** następuje zapisanie dowodu i powrót do ekranu głównego zamieszczania dowodu.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 16. Scenariusz alternatywny – po kliknięciu przycisku **Zobacz dowody** wyświetlna jest lista dowodów załączonych dla danej trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



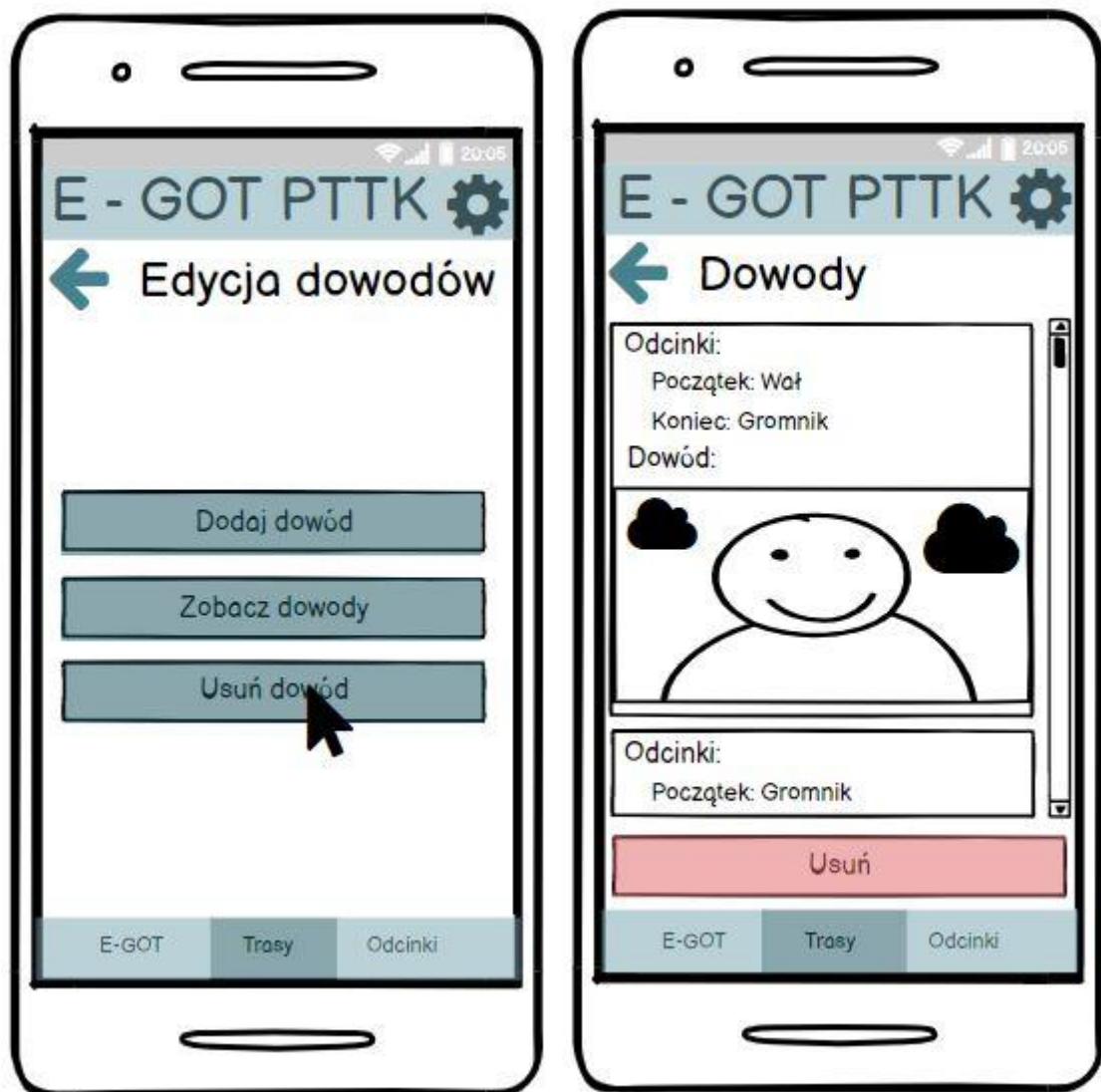
Rys 17. Scenariusz alternatywny – po kliknięciu przycisku **Zamknij** następuje powrót do ekranu głównego zamieszczania dowodów.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 18. Wyjątek – próba wyświetlenia listy dowodów załączonych dla danej trasy, jeśli nie załączono uprzednio żadnego dowodu do trasy. Aplikacja nie umożliwia wyświetlenia listy dowodów w takiej sytuacji.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 19. Scenariusz alternatywny – po kliknięciu przycisku **Usuń dowód** pojawia się ekran główny usuwania dowodu. Wyświetlana jest lista wszystkich dowodów, które zostały dodane dla trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 20. Wyjątek – próba usunięcia dowodu, jeśli nie załączono uprzednio żadnego dowodu do trasy. Aplikacja nie umożliwia usunięcia dowodu w takiej sytuacji.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 21. Scenariusz alternatywny – aby usunąć dowód należy wybrać dowód lub dowody, które mają zostać usunięte.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 22. Scenariusz alternatywny – po wybraniu dowodu i kliknięciu przycisku **Usuń** pojawi się komunikat potwierdzenia chęci usunięcia dowodu. Po naciśnięciu przycisku **OK** dowód jest usuwany, a po naciśnięciu przycisku **Cofnij** dowód nie jest usuwany i chowa się komunikat.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 23. Wyjątek – próba usunięcia dowodu bez wybrania dowodu. Aplikacja nie umożliwia usunięcia dowodu, kiedy nie wiadomo, który dowód ma zostać usunięty.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

3.2. Zapisanie trasy w książeczce



Rys 24. Scenariusz główny – widok główny tras. Po naciśnięciu przycisku **Dodaj trasę** pojawia się ekran dodania nowej trasy. W ramach extension point możliwe jest wyświetlenie szczegółów trasy, a po wyświetleniu szczegółów trasy możliwe jest zarządzanie trasą (edytacja, usuwanie – tylko jeśli trasa nie została jeszcze przekazana do potwierdzenia), dodanie dowodu oraz przekazanie jej do potwierdzenia przez przodownika.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 25. Wyjątek – próba utworzenia trasy niezawierającej żadnego odcinka.
Aplikacja nie umożliwia utworzenia takiej trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 26. Wyjątek – próba dodania dowodu do trasy niezawierającej żadnego odcinka.
Aplikacja nie umożliwia dodania dowodu w takiej sytuacji.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



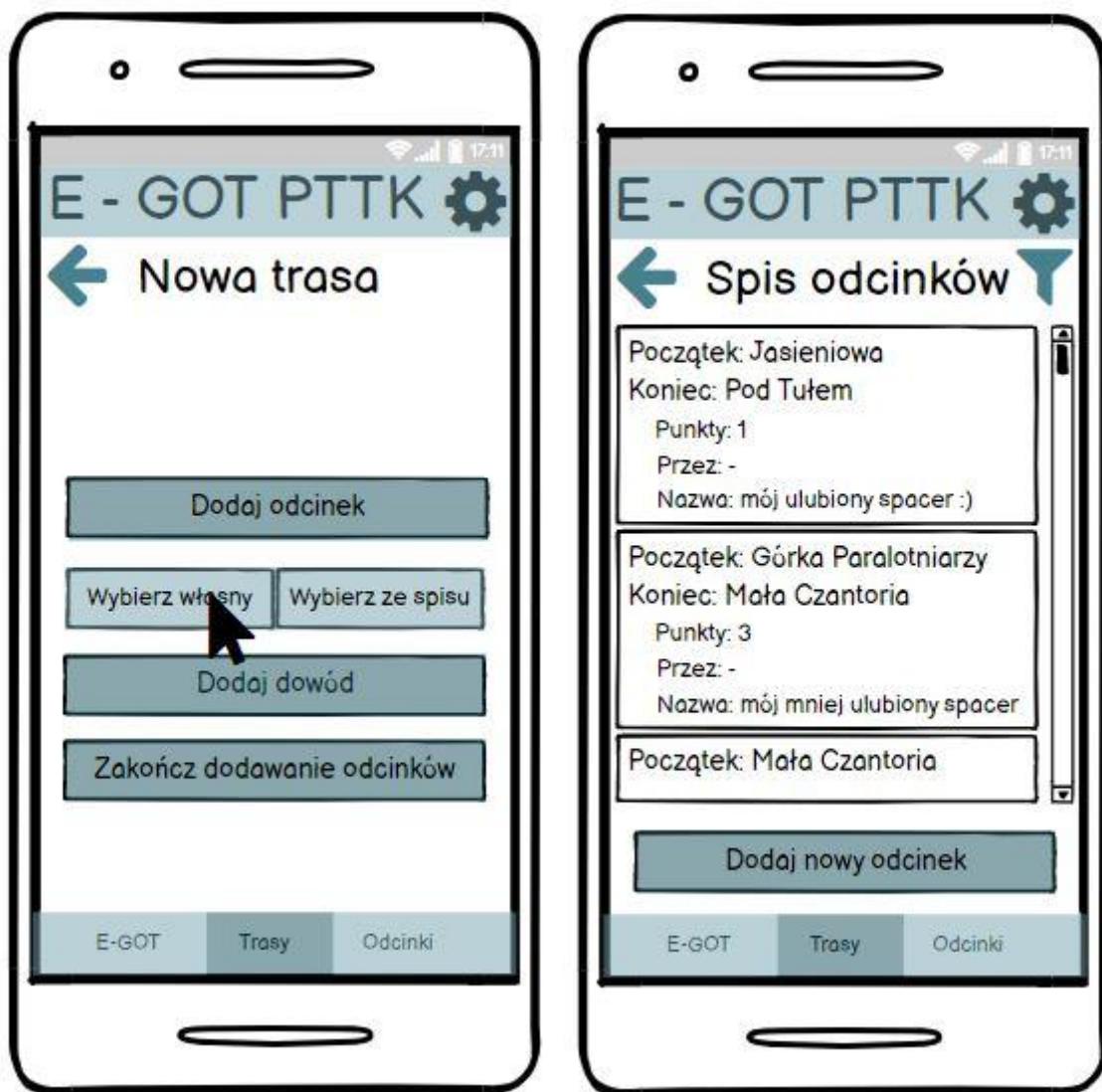
Rys 27. Scenariusz główny – po naciśnięciu przycisku **Dodaj odcinek** pojawia się możliwość wyboru własnego odcinka lub wyboru odcinka ze spisu odcinków.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 28. Scenariusz główny – po naciśnięciu **Wybierz ze spisu** wyświetlana jest lista odcinków oficjalnie punktowanych. W ramach extension point możliwe jest filtrowanie listy odcinków.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 29. Scenariusz alternatywny – po naciśnięciu **Wybierz własny** wyświetlana jest lista odcinków własnych Turysty. W ramach extension point możliwe jest filtrowanie listy odcinków.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 30. Scenariusz główny – po wybraniu odcinka pojawia się ekran umożliwiający podanie czasu przejścia odcinka i dodanie odcinka do trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



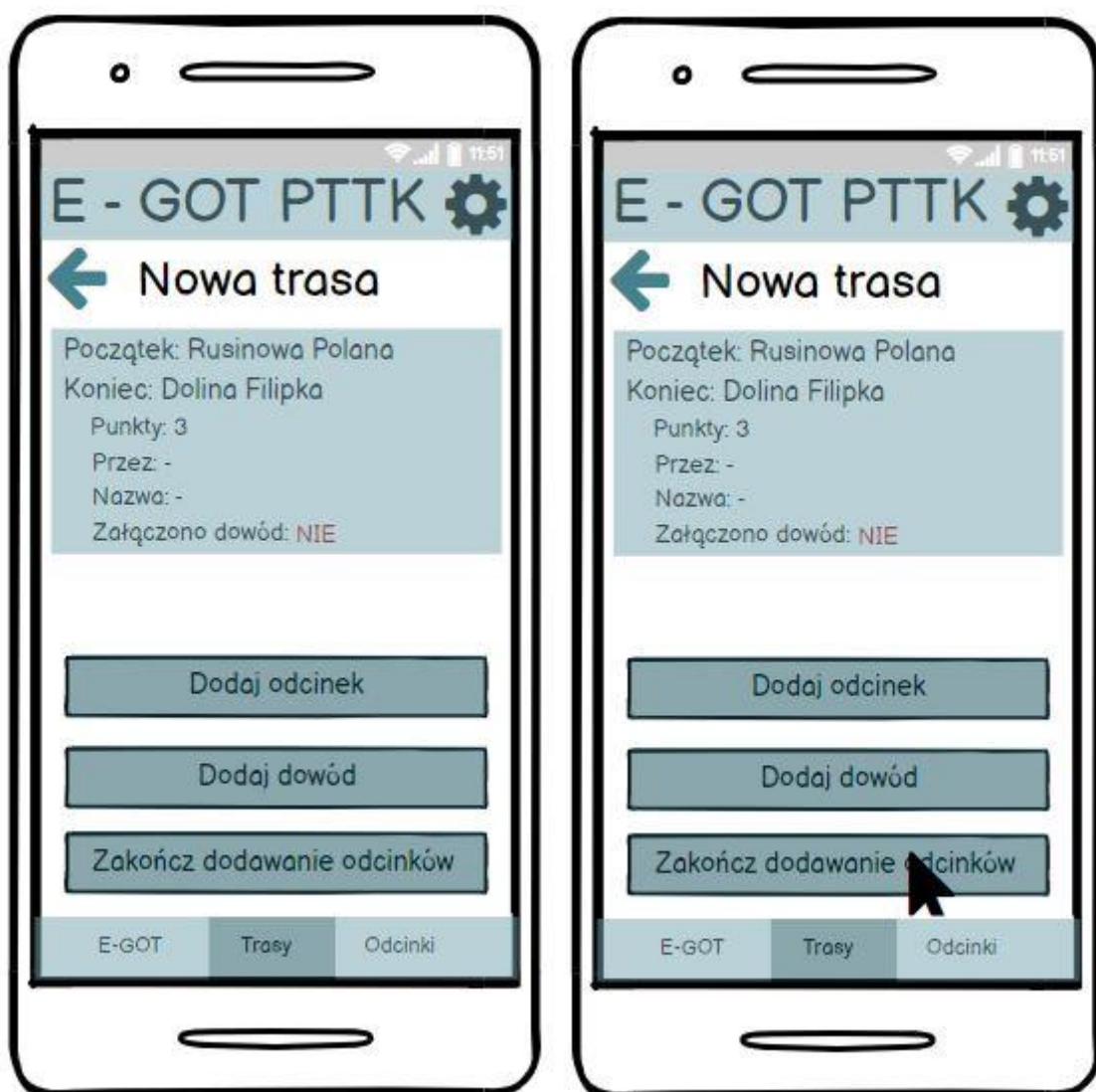
Rys 31. Wyjątek – próba dodania odcinka o zerowym lub ujemnym czasie przejścia.
Aplikacja nie umożliwia dodania takiego odcinka.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 32. Wyjątek – próba dodania odcinka, którego początek jest różny niż koniec poprzedniego. Aplikacja nie umożliwia dodania takiego odcinka, ponieważ Turyści nie mogą się teleportować :)

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



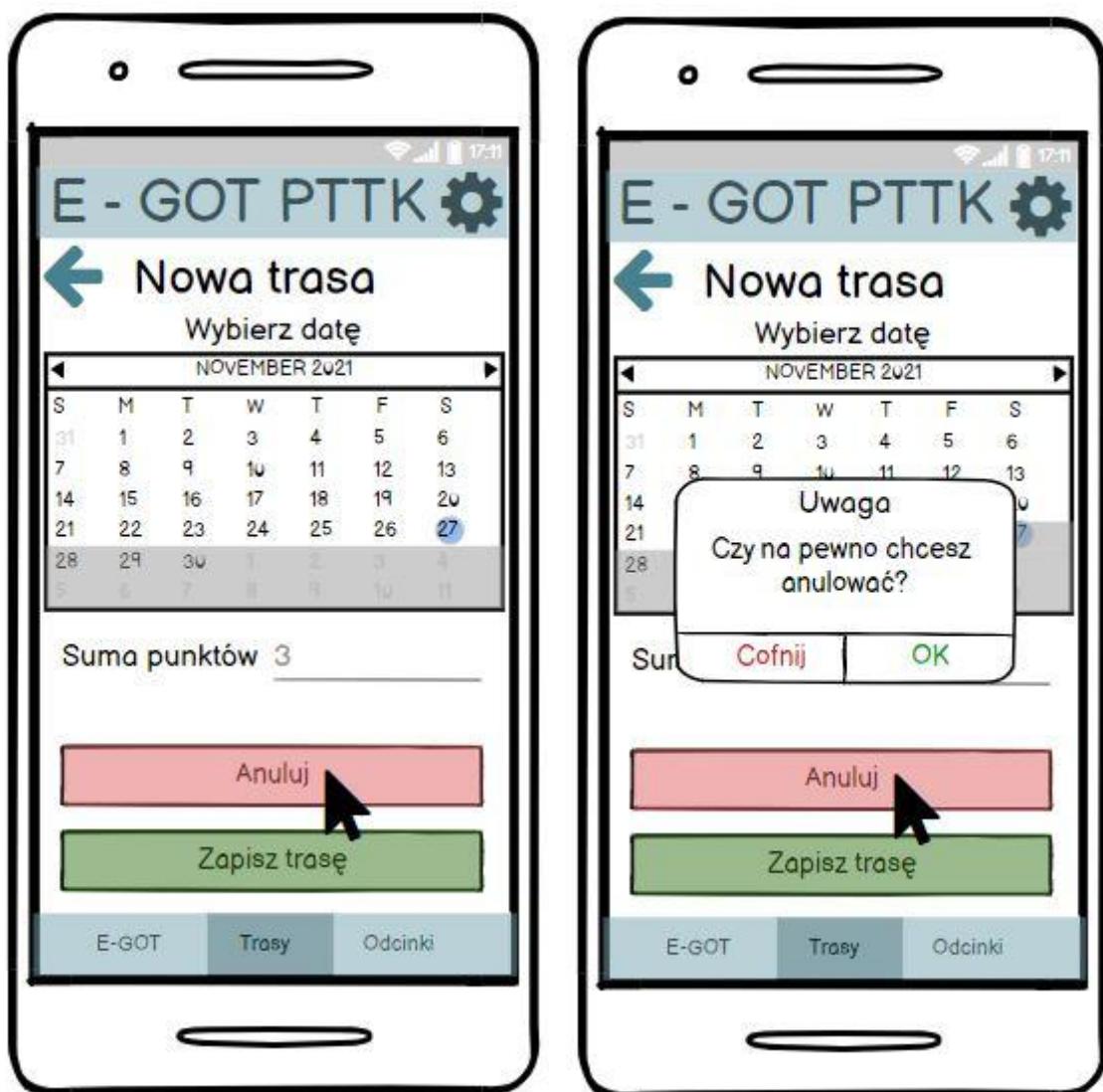
Rys 33. Scenariusz główny – po dodaniu odcinka do trasy pojawia się on na ekranie dodania nowej trasy. W ramach extension point możliwe jest dodanie dowodu potwierdzającego przebycie jednego lub kilku odcinków.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 34. Scenariusz główny – po naciśnięciu przycisku **Zakończ dodawanie odcinków** wyświetlany jest ekran wyboru daty przebycia trasy. W ramach extension point wyświetlnana jest wyliczona przez system liczba punktów za trasę.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

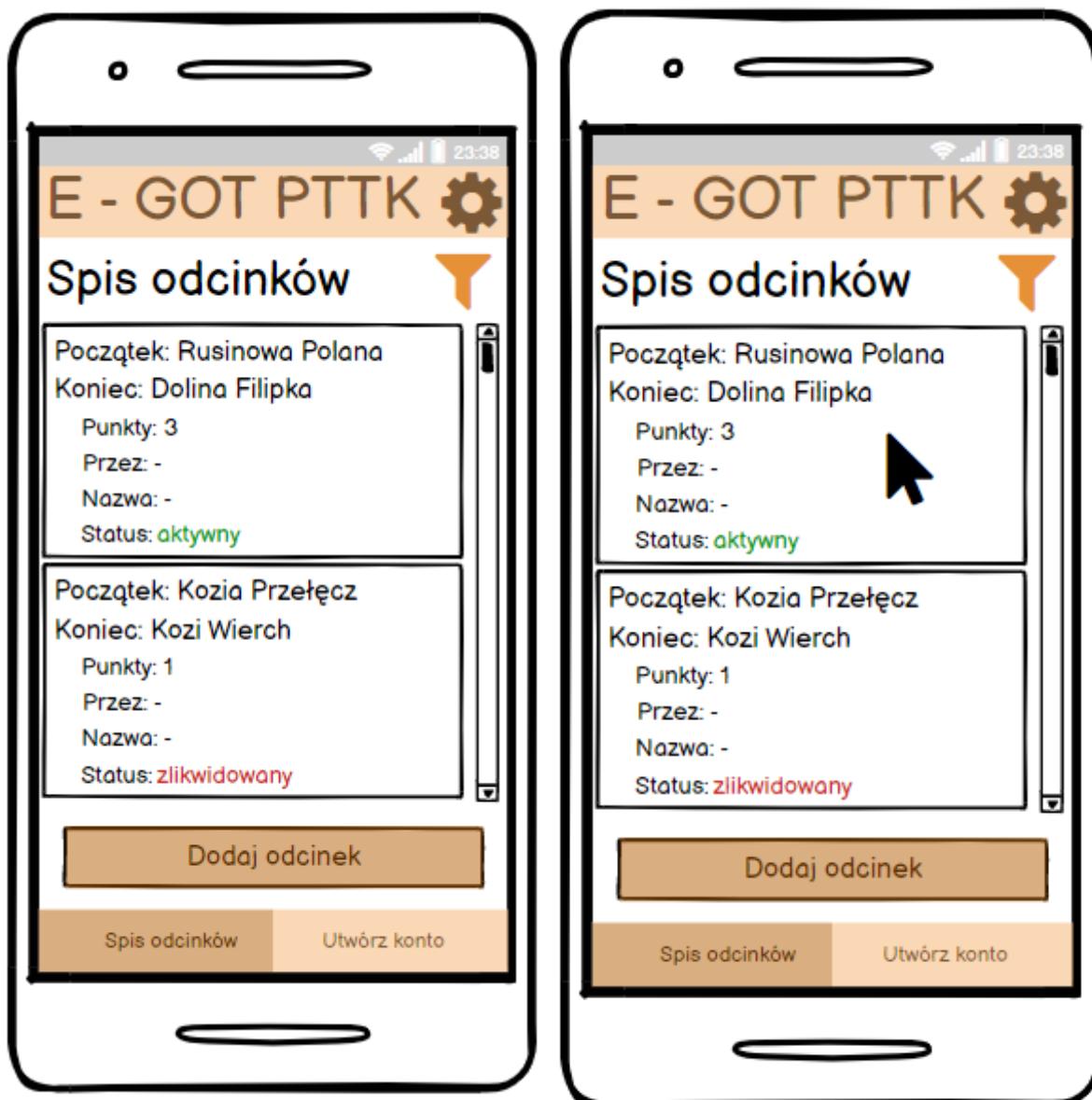


Rys 35. Wyjątek – naciśnięcie przycisku **Anuluj**. Po naciśnięciu przycisku **Anuluj** pojawia się komunikat potwierdzenia chęci anulowania. Po naciśnięciu przycisku **OK** dodawanie nowej trasy jest anulowane, a po naciśnięciu przycisku **Cofnij** dodawanie nowej trasy nie jest anulowane i chowa się komunikat.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

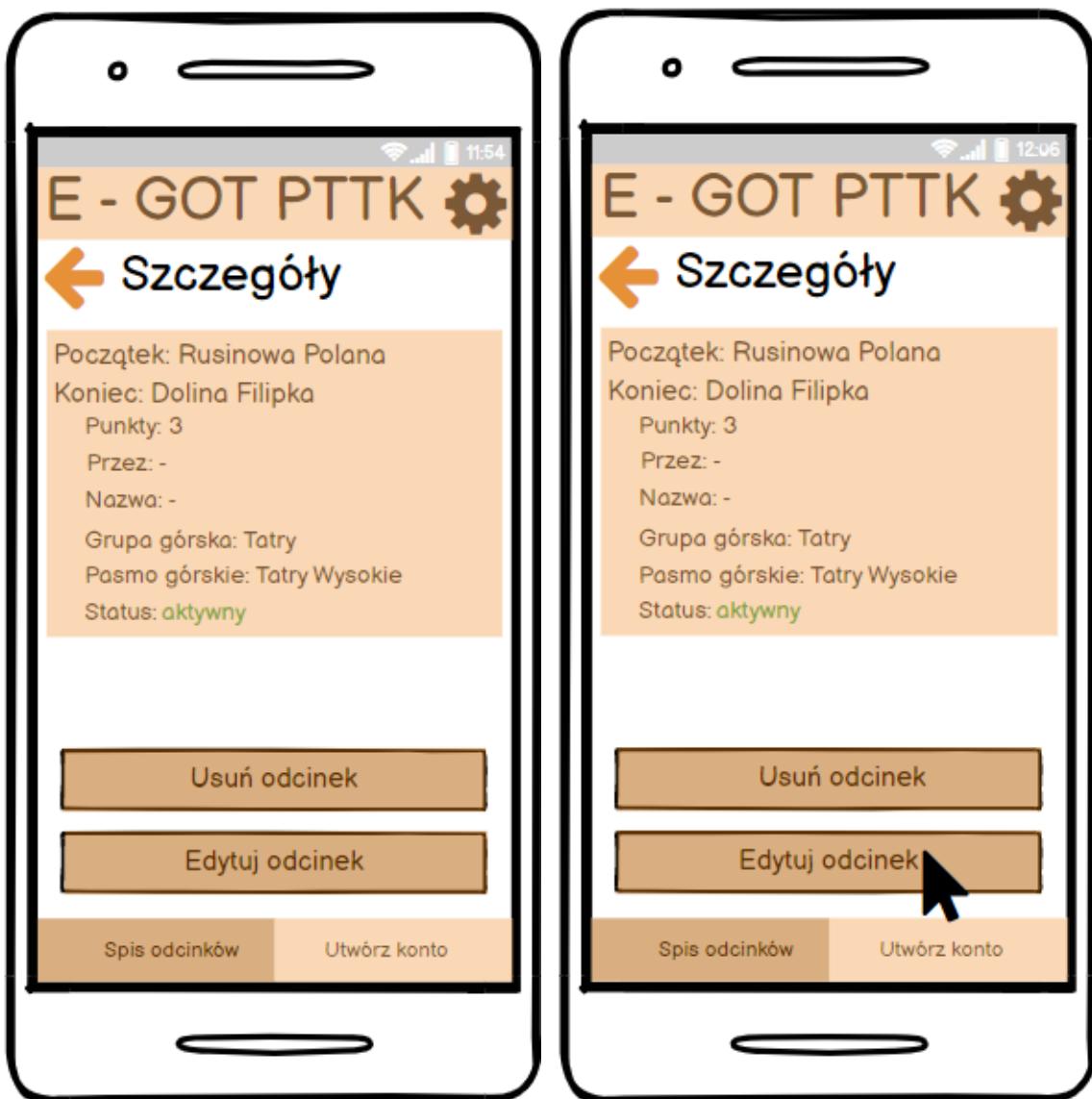
3. Prototyp interfejsu - Aleksandra Stecka

3.1. Edycja odcinka w spisie



Rys 36. Scenariusz główny – widok główny spisu odcinków. Jako extension point możliwe jest filtrowanie listy odcinków.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 37. Scenariusz główny – po wyborze odcinka wyświetlane są szczegóły odcinka. Jako extension point możliwe jest usunięcie odcinka.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 38. Scenariusz główny – po kliknięciu przycisku **Edytuj odcinek** wyświetlany jest ekran ze szczegółami odcinka i możliwością edycji.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 39. Scenariusz alternatywny – po naciśnięciu przycisku **Edytuj** przy punkcie początkowym odcinka wyświetlany jest ekran umożliwiający edycję punktu początkowego. Członek KTG PTTK może podać nazwę punktu.

Jako extension point system wyszukuje odcinki z nazwą zaczynającą się zgodnie z podanymi znakami i jeśli istnieje jakiś odcinek, do którego należy punkt o danej nazwie automatycznie uzupełniane są długość i szerokość geograficzne punktu, których nie można zmienić. Jeśli punktu o danej nazwie nie ma w żadnym odcinku zostaje utworzony nowy punkt – jego nazwę i współrzędne podaje Członek KTG PTTK – oraz system aktualizuje spis punktów o nowo utworzony punkt, a także na podstawie podanych współrzędnych lokalizuje punkt w grupie i paśmie górkim.

Analogicznie wygląda edycja punktu końcowego oraz pośredniego.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 40. Scenariusz alternatywny – po naciśnięciu przycisku **Edytuj** przy punktacji za odcinek wyświetlany jest ekran umożliwiający edycję punktacji.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 41. Wyjątek – próba podania liczby punktów mniejszej od zera. Aplikacja nie umożliwia zapisania takiej liczby punktów.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 42. Scenariusz alternatywny – po naciśnięciu przycisku **Edytuj** przy nazwie odcinka wyświetlany jest ekran umożliwiający edycję nazwy odcinka.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



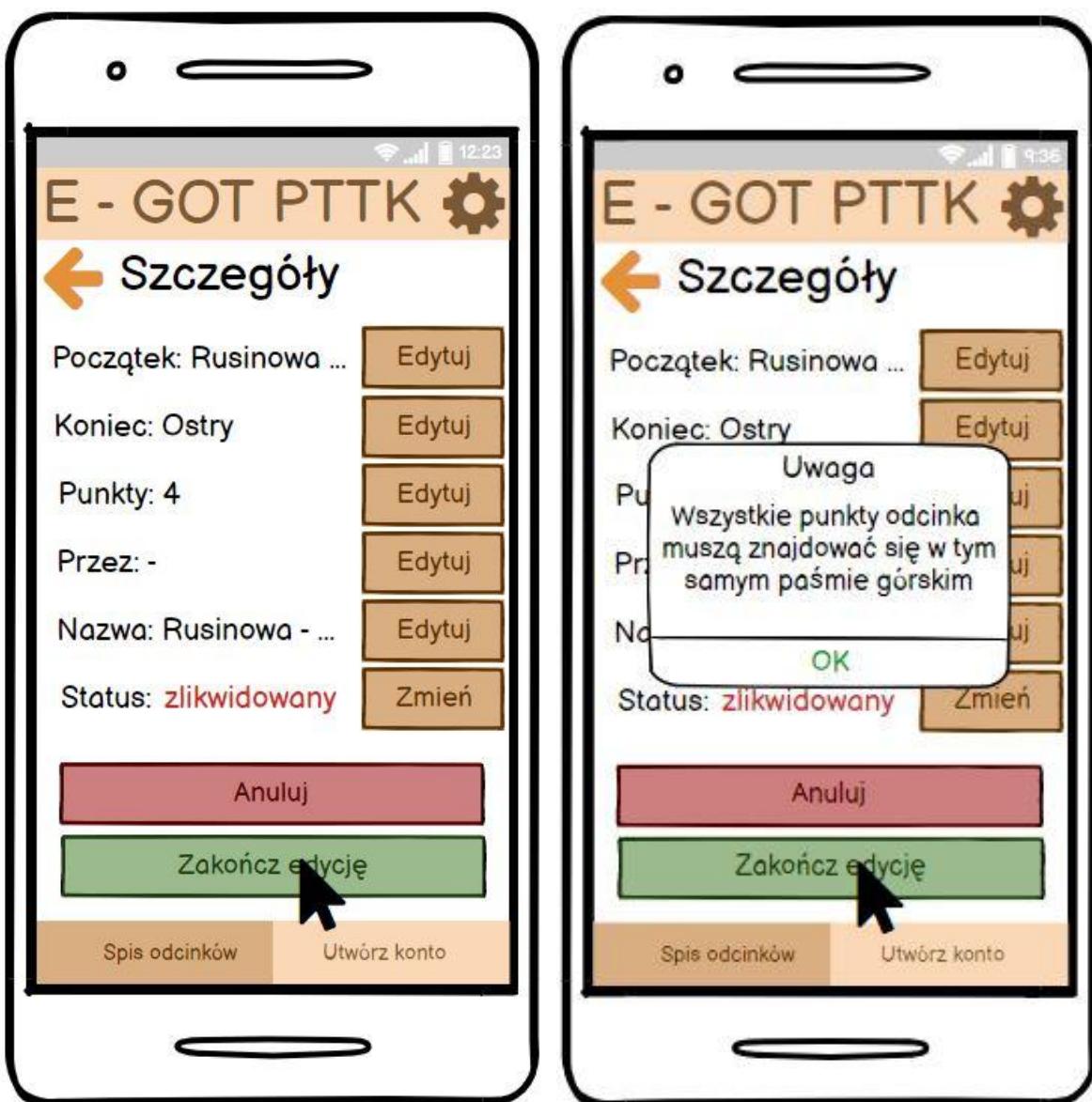
Rys 43. Scenariusz alternatywny – po naciśnięciu przycisku **Edytuj** przy statusie odcinka pojawia się komunikat potwierdzenia chęci zmiany statusu odcinka. Po naciśnięciu przycisku **OK** zapisany zostaje nowy status odcinka, a po naciśnięciu przycisku **Cofnij** status odcinka nie zmienia się i chowa się komunikat.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 44. Scenariusz alternatywny – po naciśnięciu przycisku **OK** zapisany zostaje nowy status odcinka.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 45. Wyjątek – próba utworzenia odcinka, którego punkty początkowy, końcowy i pośredni znajdują się w różnych grupach górskich. Aplikacja nie umożliwia utworzenia takiego odcinka.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 46. Wyjątek – naciśnięcie przycisku **Anuluj**. Po naciśnięciu przycisku **Anuluj** pojawia się komunikat potwierdzenia chęci anulowania. Po naciśnięciu przycisku **OK** dodawanie nowej trasy jest anulowane, a po naciśnięciu przycisku **Cofnij** dodawanie nowej trasy nie jest anulowane i chowa się komunikat.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

3.2. Potwierdzenie trasy



Rys 47. Scenariusz główny – widok główny potwierdzenia trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



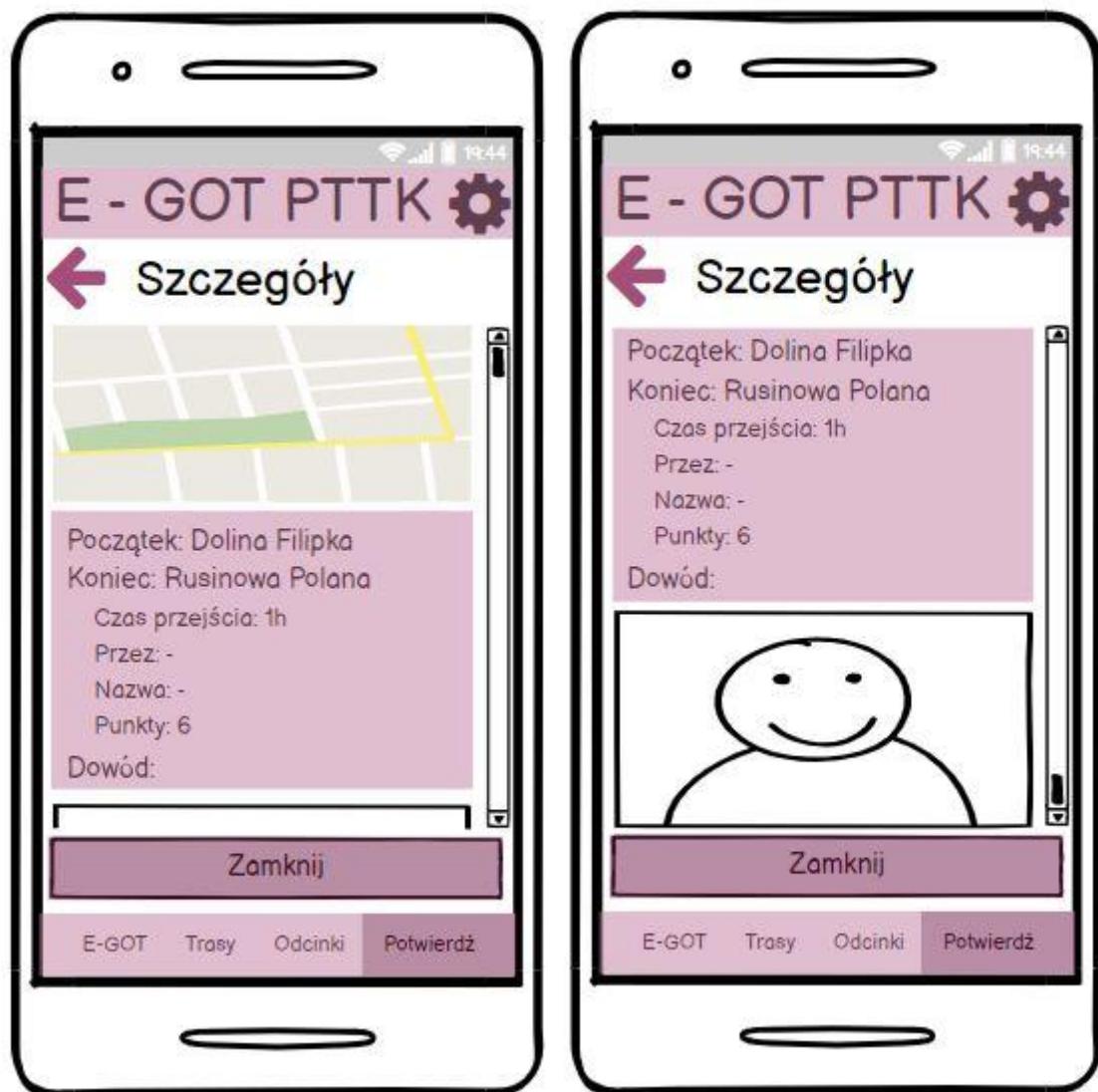
Rys 48. Scenariusz alternatywny – po naciśnięciu przycisku **Pokaż odcinki** wyświetlana jest lista wszystkich odcinków trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 49. Scenariusz alternatywny – Przodownik ma możliwość wybrania konkretnego odcinka z listy odcinków trasy w celu zobaczenia jego szczegółów.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 50. Scenariusz alternatywny – po wybraniu konkretnego odcinka wyświetlane są szczegóły danego odcinka, w tym punkty początkowy, końcowy oraz pośredni, czas przejścia, nazwa oraz liczba punktów za odcinek, odcinek zaznaczany jest na mapie oraz wyświetlany jest dowód przypisany do odcinka.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



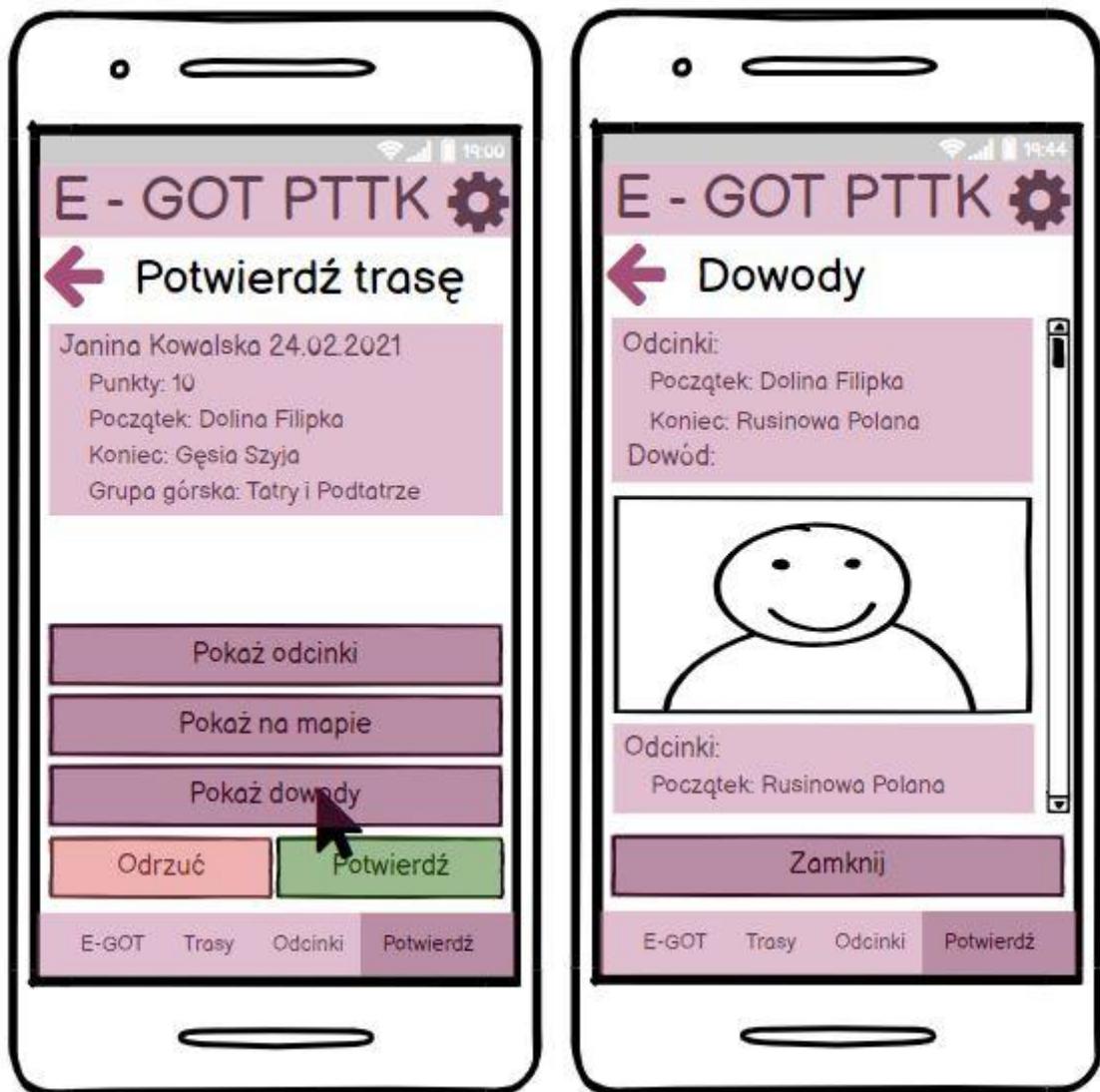
Rys 51. Scenariusz alternatywny – po kliknięciu przycisku **Zamknij** powrót do szczegółów trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



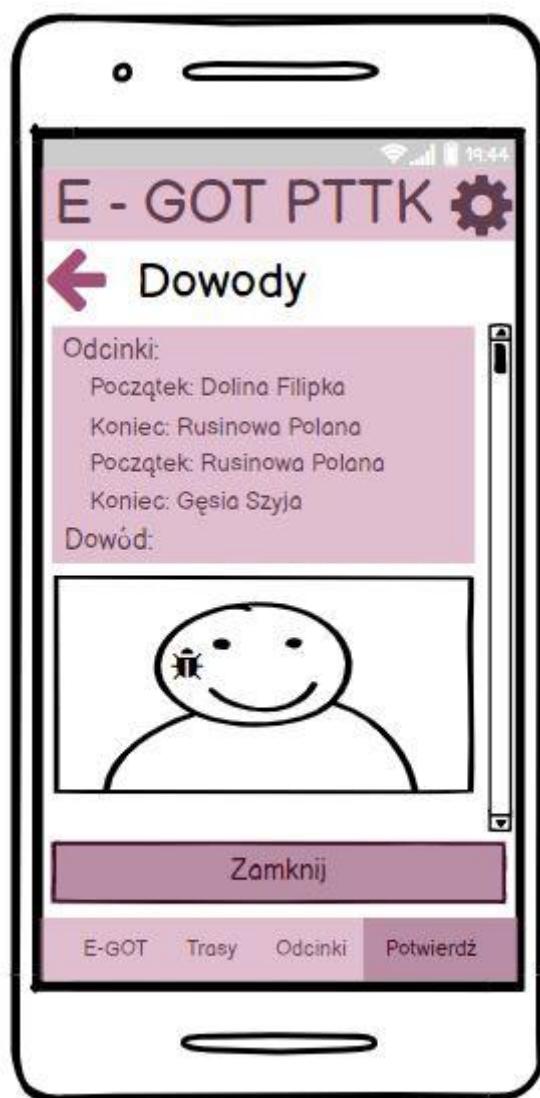
Rys 52. Scenariusz alternatywny – po naciśnięciu przycisku **Pokaż na mapie** uruchamiana jest aplikacja do przeglądania map (na przykład Google Maps), na której zaznaczona jest trasa.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



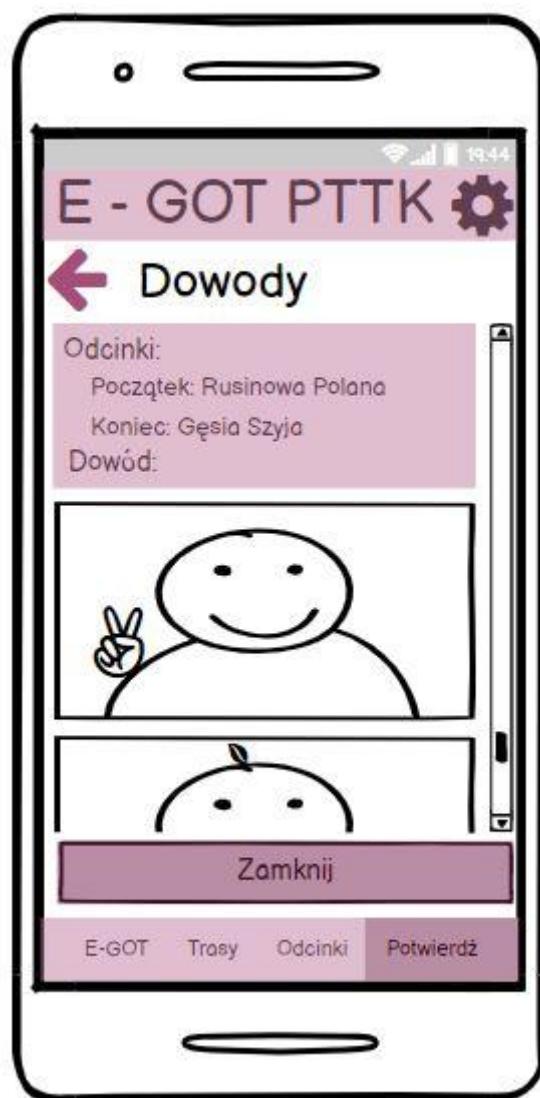
Rys 53. Scenariusz alternatywny – po naciśnięciu przycisku **Pokaż dowody** wyświetlana jest lista dowodów wraz z informacją, do którego odcinka zostały przypisane.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



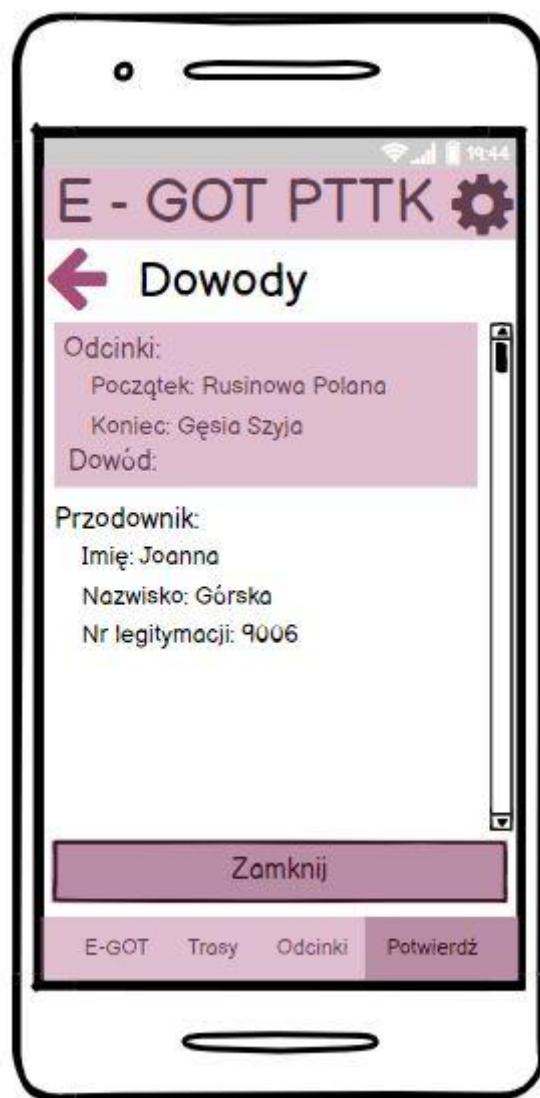
Rys 54. Scenariusz alternatywny – wariant wyświetlania listy dowodów, w którym jeden dowód potwierdza przebycie wielu odcinków.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



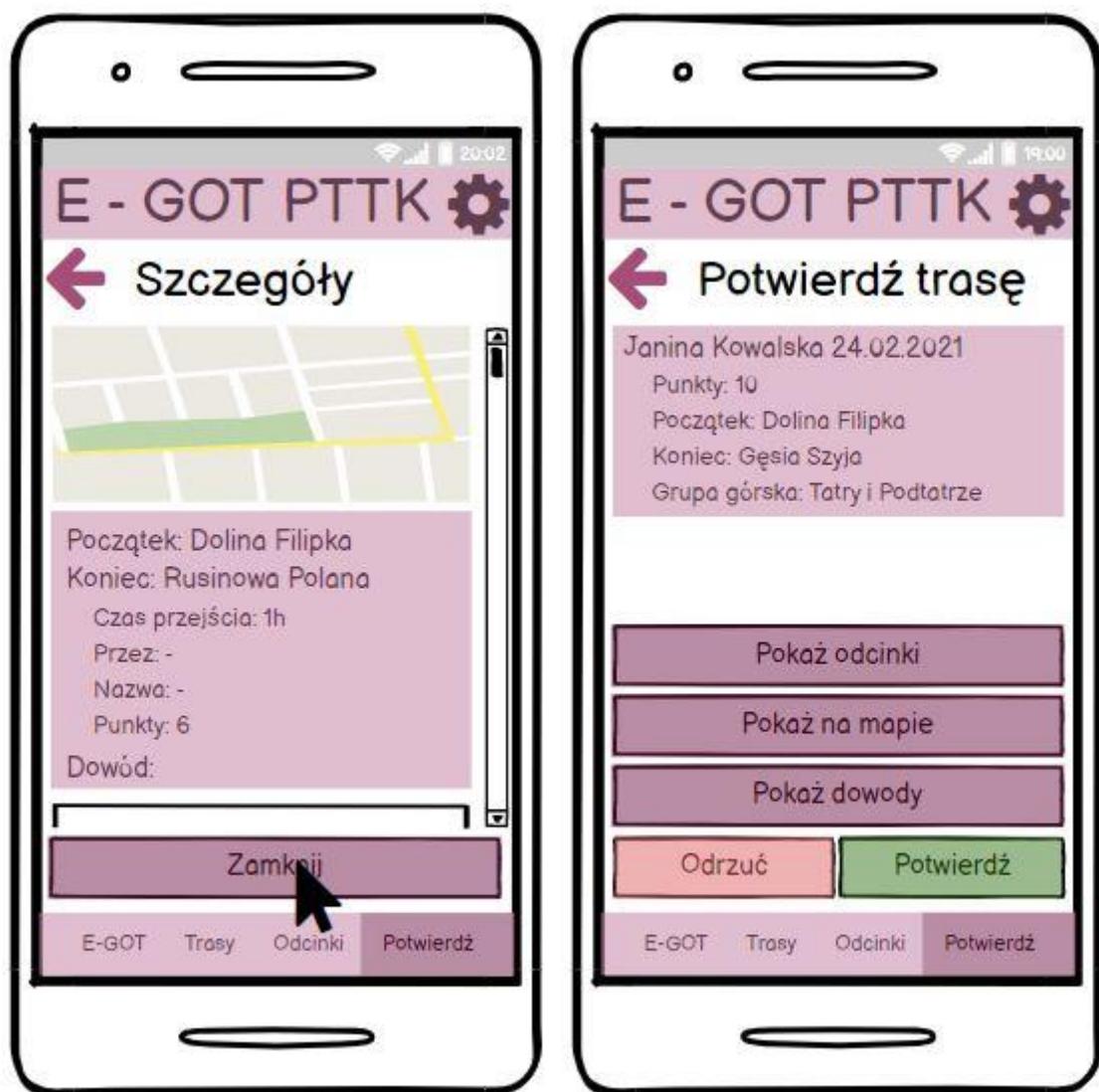
Rys 55. Scenariusz alternatywny – wariant wyświetlania listy dowodów, w którym dla jednego odcinka dodano kilka dowodów.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 56. Scenariusz alternatywny – wariant wyświetlania listy dowodów, w którym przodownik uczestniczył w przebywaniu odcinka i jako dowód zostały dodane jego dane.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 57. Scenariusz alternatywny – po kliknięciu przycisku **Zamknij** następuje powrót do szczegółów trasy.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 58. Scenariusz główny – potwierdzenie trasy naciśnięciem przycisku **Potwierdź**. Po naciśnięciu przycisku **Potwierdź** pojawia się komunikat potwierdzenia chęci potwierdzenia trasy. Po naciśnięciu przycisku **OK** status trasy zmienia się na “zatwierdzona”, a po naciśnięciu przycisku **Cofnij** potwierdzanie trasy zostaje anulowane i chowa się komunikat.

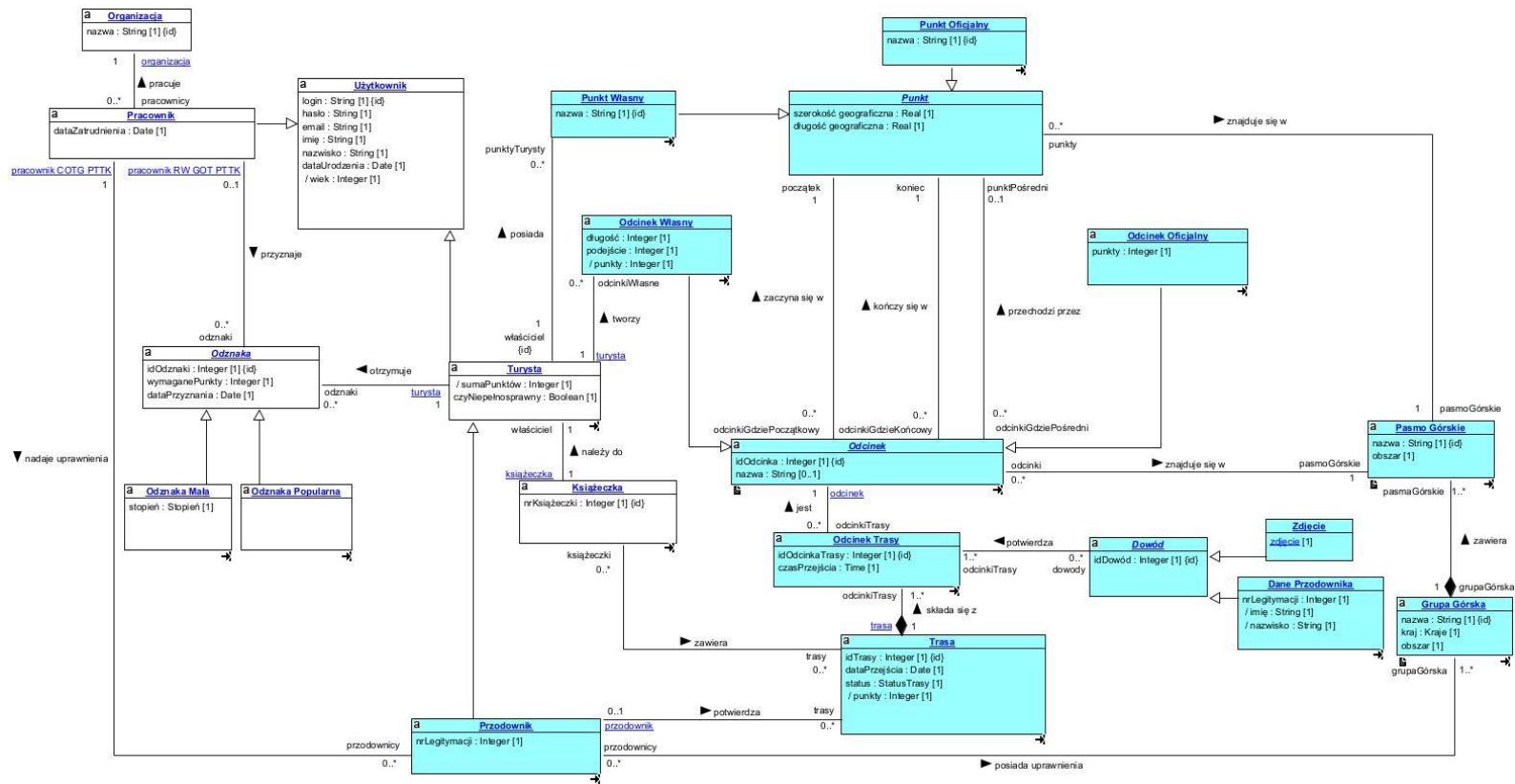
E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.



Rys 59. Scenariusz alternatywny – odrzucenie trasy naciśnięciem przycisku **Odrzuć**.

Po naciśnięciu przycisku **Odrzuć** pojawia się komunikat potwierdzenia chęci odrzucenia trasy. Po naciśnięciu przycisku **OK** status trasy zmienia się na "niezatwierdzona", a po naciśnięciu przycisku **Cofnij** odrzucanie trasy zostaje anulowane i chowa się komunikat.

Model informacyjny



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Wyniki etapu 3: Architektura systemu, projekt bazy danych, realizacja przypadków użycia

E - książeczka GOT PTTK

Projektowanie oprogramowania



E - GOT PTTK

Skład zespołu:
Agnieszka Kłobus
Aleksandra Stecka
(zespół 8)

Prowadzący:
dr inż. Zbigniew Szpunar

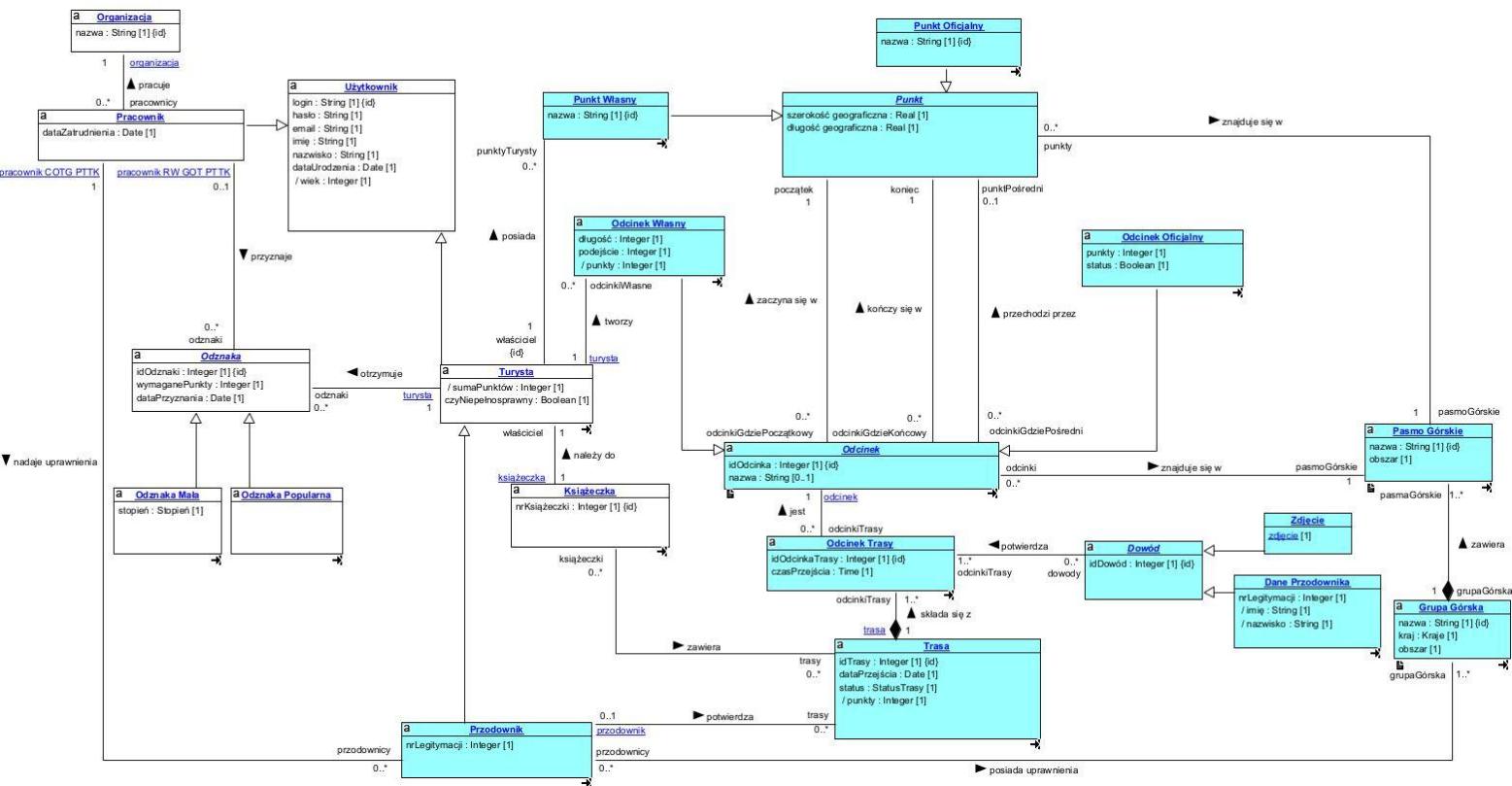
E-księżeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Zmiany do Etapu 1:

- zmieniono wymaganie "Możliwość uruchomienia za pomocą przeglądarki internetowej" na "Możliwość uruchomienia za pomocą aplikacji mobilnej" (punkt 5 Etapu 1) – już w etapie 2 podjęto decyzję o przygotowaniu aplikacji mobilnej, ale nie poprawiono wtedy tego wymagania.

Zmiany do Etapu 2:

- na modelu informacyjnym dodano asocjację pomiędzy Punktem a Pasmem Górkim – *W paśmie górkim może znajdować się zero lub wiele punktów. Punkt musi znajdować się w dokładnie jednym paśmie górkim.*
 - w modelu informacyjnym dodano atrybut Odcinka Oficjalnego – status.



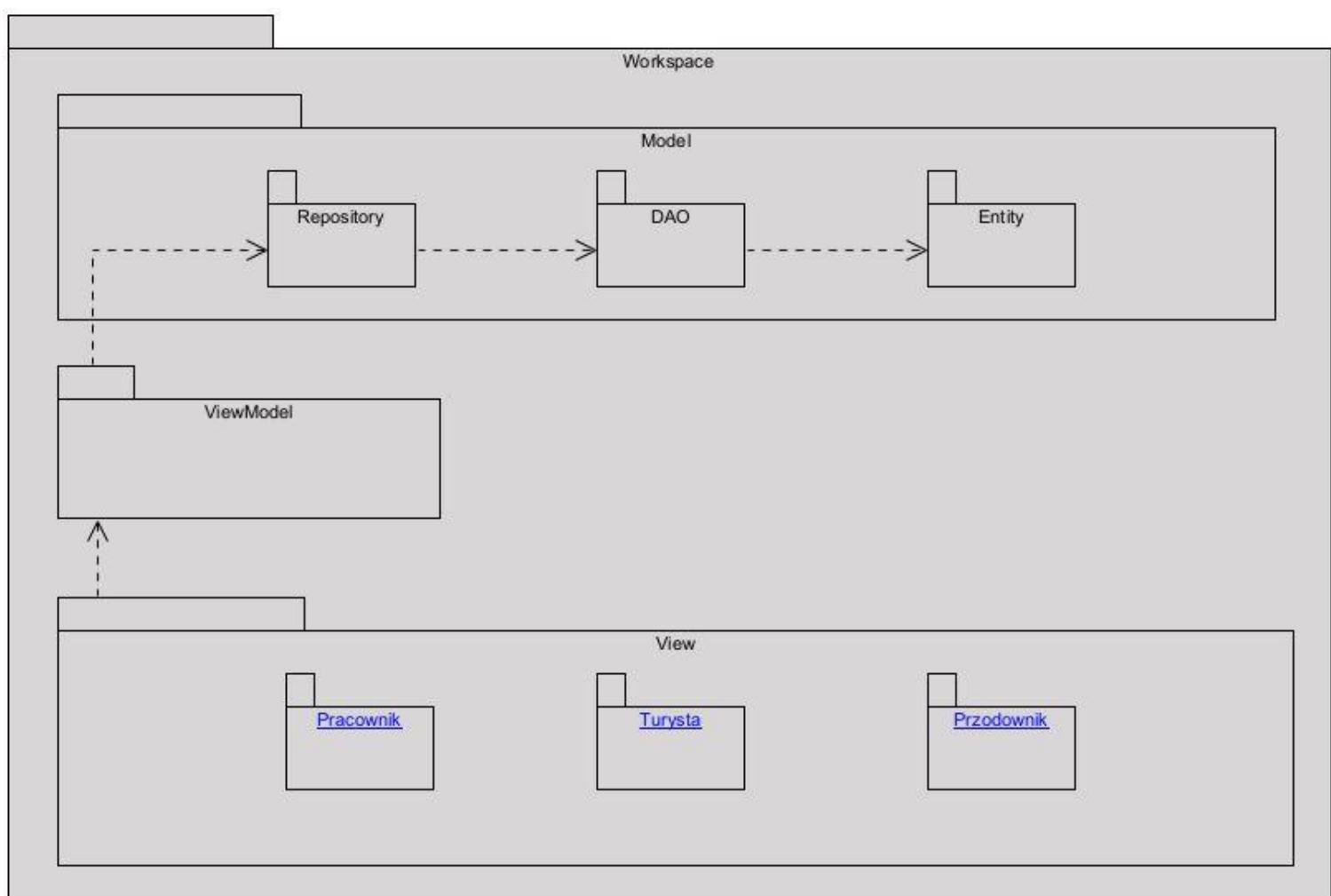
Poprzednio zakładano, że podczas dodawania odcinka bez wiedzy użytkownika zostają utworzone nowe punkty lub, jeśli istnieją one w spisie punktów oficjalnych lub własnych, zostają wybrane istniejące punkty z odpowiedniego spisu. Punkty te są wtedy na podstawie ich współrzędnych lokalizowane, a pasmo górskie do którego należą zostaje zapisane jako własność odcinka.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

W procesie projektowania bazy danych uznano jednak, że dla skutecznego jej działania wskazane jest przechowywanie informacji o paśmie górkim także jako atrybutu punktu.

Architektura oprogramowania

1. Architektura logiczna



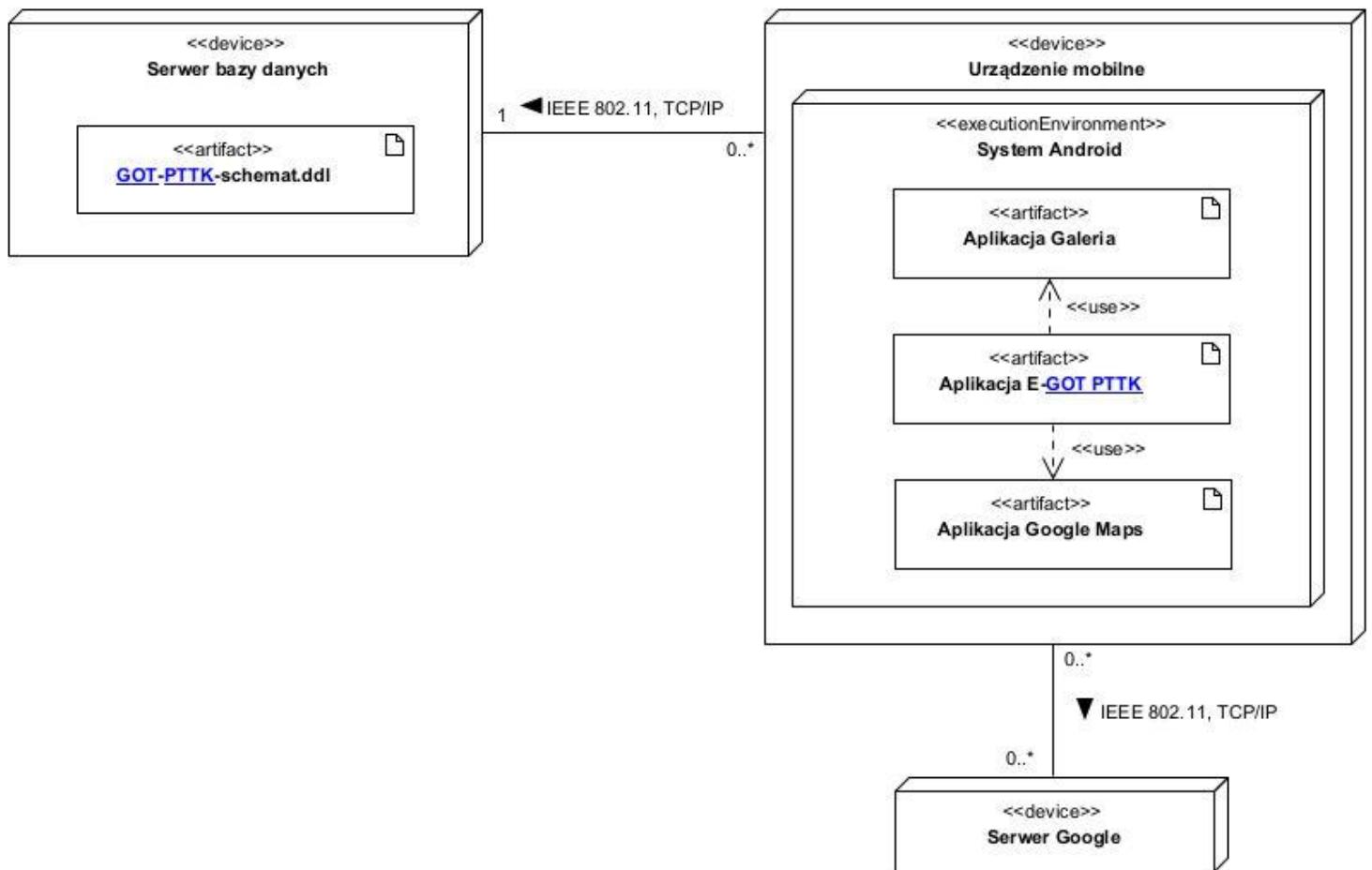
Nazwa	Opis
Workspace	Pakiet zawierający wszystkie pliki systemu.
Model	Warstwa logiki wykorzystywana we wzorcu architektury MVVM. Odpowiada za połaczanie aplikacji do bazy danych i dostęp do danych.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

ViewModel	Warstwa aplikacji wykorzystywana we wzorcu architektury MVVM. Odpowiedzialny za odczytywanie informacji z Modelu i przekazywanie ich do View, w celu ich wyświetlenia.
View	Warstwa interfejsu wykorzystywana we wzorcu architektury MVVM. Warstwa "najbliżej" użytkownika, udostępniająca mu interfejs graficzny aplikacji.
DAO	Pakiet grupujący interfejsy dostępu do danych DAO – Data Access Object.
Entity	Pakiet grupujący klasy określające strukturę danych w bazie.
Repository	Pakiet grupujący klasy odpowiedzialne za dostęp do danych z bazy.
Pracownik	Pakiet grupujący wszystkie pliki XML tworzące interfejs widziany przez Pracownika.
Turysta	Pakiet grupujący wszystkie pliki XML tworzące interfejs widziany przez Turystę.
Przodownik	Pakiet grupujący wszystkie pliki XML tworzące interfejs widziany przez Przodownika.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

2. Architektura fizyczna

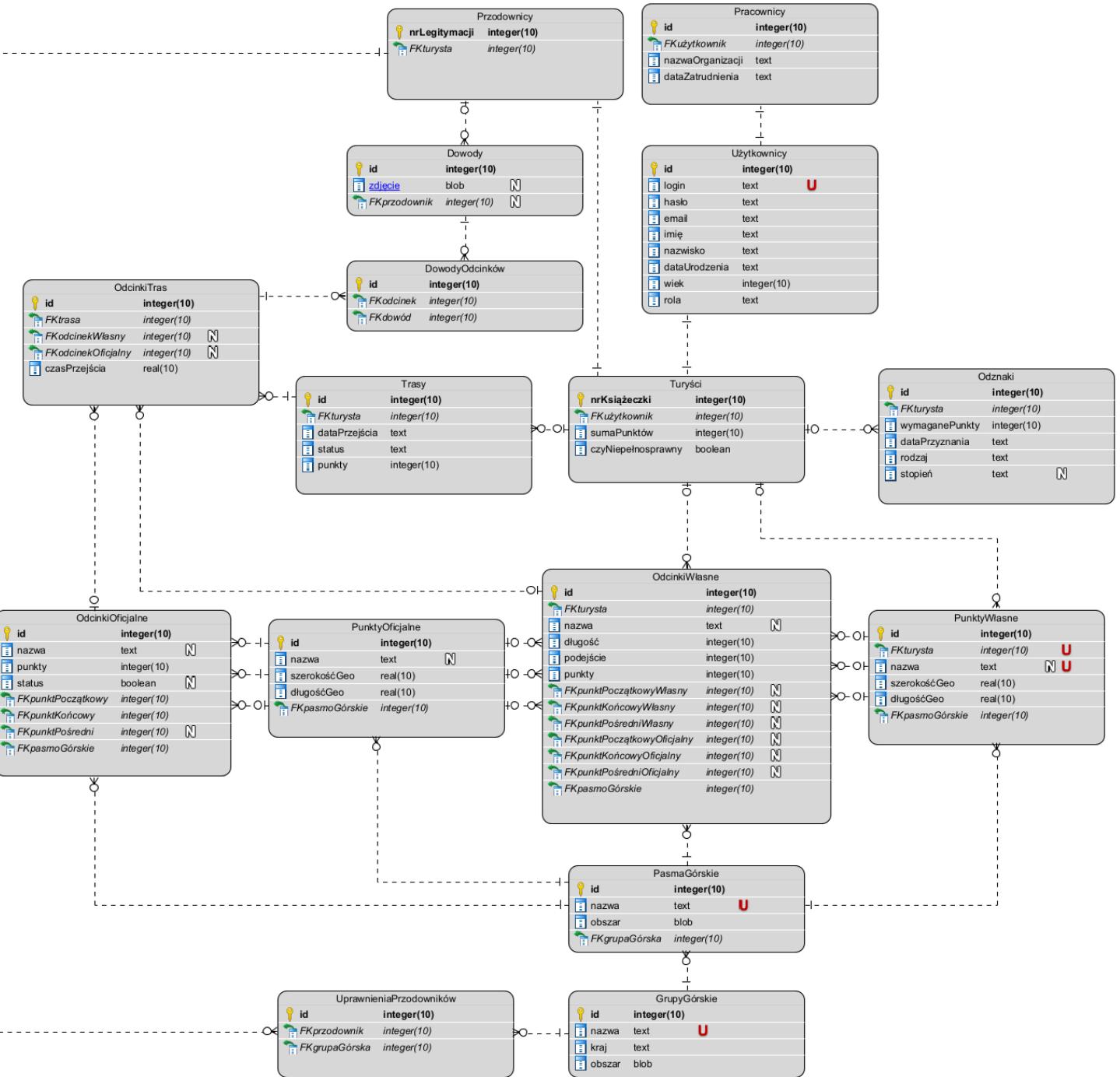


E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Nazwa	Opis
Serwer bazy danych	Urządzenie fizyczne świadczące usługi udostępniania zasobów znajdujących się w bazie danych. Serwer udostępnia urządzeniom mobilnym przechowywane w bazie danych dane aplikacji.
Urządzenie mobilne	Urządzenie mobilne należące do konkretnego użytkownika aplikacji. Urządzenie musi mieć system operacyjny Android, ponieważ na ten system przeznaczona jest aplikacja opracowywana w ramach projektu.
System Android	System operacyjny urządzeń mobilnych. Na nim zainstalowana i uruchamiana jest aplikacja E-GOT PTTK.
Serwer Google	Serwer firmy Google przechowujący dane aplikacji tej firmy. Przechowuje m.in. dane dotyczące map satelitarnych, tras oraz widoków panoramicznych 360°.
GOT-PTTK-schemat.ddl	Relacyjna baza danych przechowująca dane aplikacji oraz powiązania między tymi danymi.
Aplikacja E-GOT PTTK	Aplikacja mobilna opracowywana w ramach projektu.
Aplikacja Galeria	Systemowa aplikacja mobilna umożliwiająca przeglądanie plików graficznych na urządzeniu mobilnym.
Aplikacja Google Maps	Aplikacja mobilna firmy Google umożliwiająca dostęp do lokalizacji oraz prezentowania lokalizacji na mapie świata.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

3. Model danych



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

4. Realizacja przypadków użycia

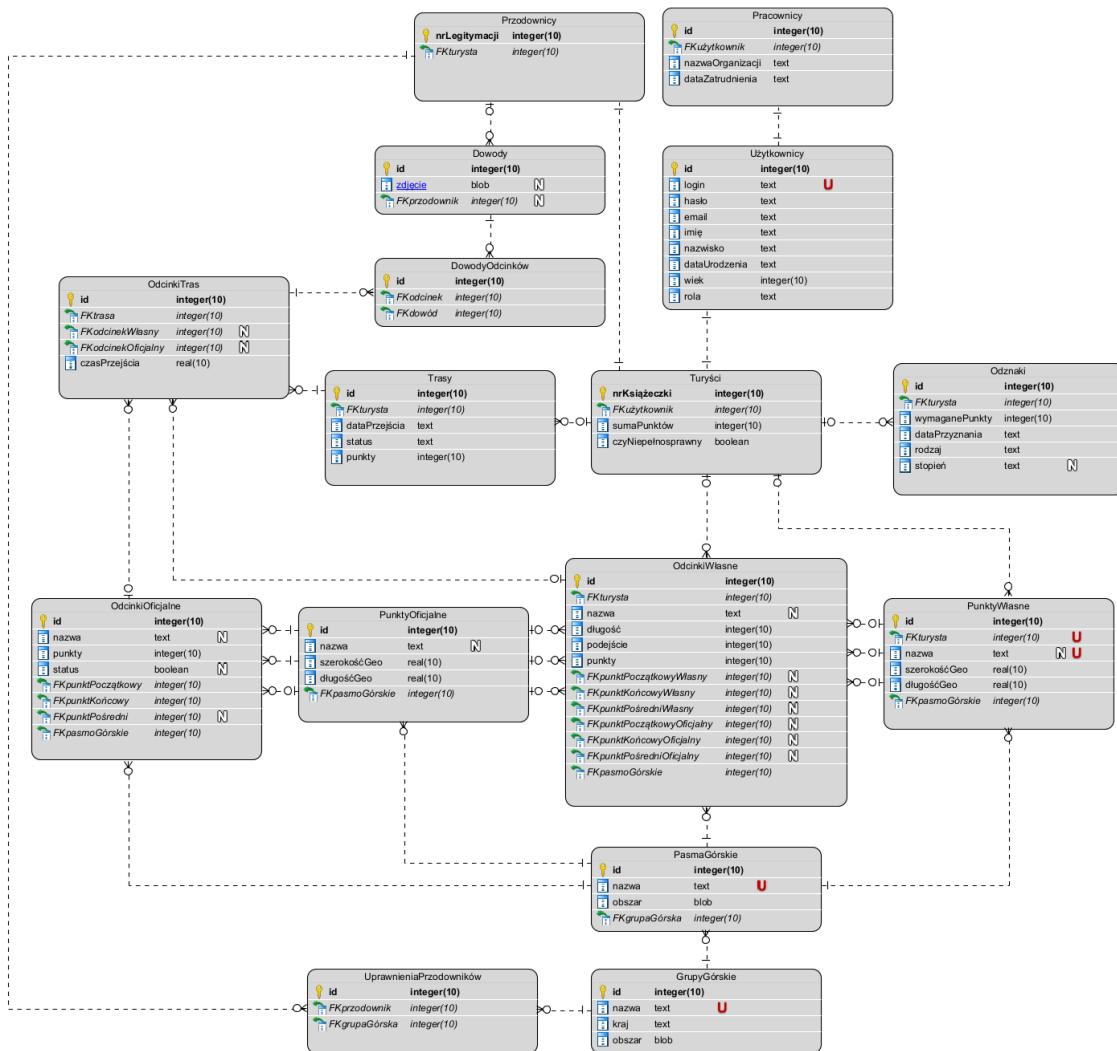
Diagramy sekwencji i diagramy klas zostaną wykonane po implementacji.

Projekt bazy danych

Typ bazy danych: relacyjna

System zarządzania bazą danych: SQLite

Struktura bazy danych:



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Skrypt SQL DDL

wygenerowany za pomocą Visual Paradigm na podstawie diagramu encji

```

CREATE TABLE Użytkownicy (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    login       text NOT NULL UNIQUE,
    hasło       text NOT NULL,
    email       text NOT NULL,
    imię        text NOT NULL,
    nazwisko    text NOT NULL,
    dataUrodzenia text NOT NULL,
    wiek        integer(10) NOT NULL,
    rola         Rola DEFAULT 'turysta' NOT NULL CREATE TYPE Rola AS
ENUM('turysta','przodownik','pracownik','członek'));

CREATE TABLE Turyści (
    nrKsiążeczki      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKużytkownik      integer(10) NOT NULL,
    sumaPunktów       integer(10) DEFAULT 0 NOT NULL,
    czyNiepełnosprawny boolean NOT NULL,
    FOREIGN KEY(FKużytkownik) REFERENCES Użytkownicy(id));

CREATE TABLE Przodownicy (
    nrLegitymacji INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKturysta      integer(10) NOT NULL,
    FOREIGN KEY(FKturysta) REFERENCES Turyści(nrKsiążeczki));

CREATE TABLE Pracownicy (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKużytkownik      integer(10) NOT NULL,
    nazwaOrganizacji Organizacja NOT NULL CREATE TYPE Organizacja AS ENUM('COTG
PTTK','RW GOT PTTK'),
    dataZatrudnienia text NOT NULL,
    FOREIGN KEY(FKużytkownik) REFERENCES Użytkownicy(id));

CREATE TABLE Trasy (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKturysta      integer(10) NOT NULL,
    dataPrzejścia text NOT NULL,
    status        StatusTrasy DEFAULT 'oczekuje na wysłanie' NOT NULL CREATE
TYPE StatusTrasy AS ENUM('zatwierdzona','niezatwierdzona','oczekuje na
wysłanie','oczekuje na potwierdzenie'),
    punkty        integer(10) NOT NULL,
    FOREIGN KEY(FKturysta) REFERENCES Turyści(nrKsiążeczki));

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

CREATE TABLE OdcinkiTras (
    id                  INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKtrasa            integer(10) NOT NULL,
    FKodcinekWłasny   integer(10),
    FKodcinekOficjalny integer(10),
    czasPrzejścia     real(10) NOT NULL,
    FOREIGN KEY(FKtrasa) REFERENCES Trasy(id),
    FOREIGN KEY(FKodcinekOficjalny) REFERENCES OdcinkiOficjalne(id),
    FOREIGN KEY(FKodcinekWłasny) REFERENCES OdcinkiWłasne(id),
    CONSTRAINT własnyXORoficjalny
        CHECK ((OdcinkiTras.FKodcinekWłasny IS NULL OR
OdcinkiTras.FKodcinekOficjalny IS NULL) AND (OdcinkiTras.FKodcinekWłasny IS
NOT NULL OR OdcinkiTras.FKodcinekOficjalny IS NOT NULL)));

```



```

CREATE TABLE OdcinkiWłasne (
    id                  INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKturysta          integer(10) NOT NULL,
    nazwa               text,
    długość             integer(10) NOT NULL,
    podejście           integer(10) NOT NULL,
    punkty              integer(10) NOT NULL,
    FKpunktPoczątkowyWłasny integer(10),
    FKpunktKońcowyWłasny integer(10),
    FKpunktPośredniWłasny integer(10),
    FKpunktPoczątkowyOficjalny integer(10),
    FKpunktKońcowyOficjalny integer(10),
    FKpunktPośredniOficjalny integer(10),
    FKpasmogórskie      integer(10) NOT NULL,
    FOREIGN KEY(FKturysta) REFERENCES Turyści(nrKsiążeczk),
    FOREIGN KEY(FKpunktPoczątkowyWłasny) REFERENCES PunktyWłasne(id),
    FOREIGN KEY(FKpunktKońcowyWłasny) REFERENCES PunktyWłasne(id),
    FOREIGN KEY(FKpunktPoczątkowyOficjalny) REFERENCES PunktyOficjalne(id),
    FOREIGN KEY(FKpunktKońcowyOficjalny) REFERENCES PunktyOficjalne(id),
    FOREIGN KEY(FKpunktPośredniOficjalny) REFERENCES PunktyOficjalne(id),
    FOREIGN KEY(FKpunktPośredniWłasny) REFERENCES PunktyWłasne(id),
    FOREIGN KEY(FKpasmogórskie) REFERENCES PasmaGórskie(id),
    CONSTRAINT POCZĄTEKwłasnyXORoficjalny
        CHECK ((OdcinkiWłasne.FKpunktPoczątkowyWłasny IS NULL OR
OdcinkiWłasne.FKpunktPoczątkowyOficjalny IS NULL) AND
(OdcinkiWłasne.FKpunktPoczątkowyWłasny IS NOT NULL) OR
OdcinkiWłasne.FKpunktPoczątkowyOficjalny IS NOT NULL)),
    CONSTRAINT KONIECwłasnyXORoficjalny
        CHECK ((OdcinkiWłasne.FKpunktKońcowyWłasny IS NULL OR
OdcinkiWłasne.FKpunktKońcowyOficjalny IS NULL) AND

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```
(OdcinkiWłasne.FKpunktKońcowyWłasny      IS      NOT      NULL      OR
OdcinkiWłasne.FKpunktKońcowyOficjalny IS NOT NULL)) ,
CONSTRAINT POŚREDNIWłasnyNANDoficjalny
    CHECK  (OdcinkiWłasne.FKpunktPośredniWłasny   IS      NOT      NULL      OR
OdcinkiWłasne.FKpunktPośredniOficjalny IS NOT NULL));
```

```
CREATE TABLE OdcinkiOficjalne (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    nazwa       text,
    punkty     integer(10) DEFAULT 0 NOT NULL,
    FKpunktPoczątkowy integer(10) NOT NULL,
    FKpunktKońcowy   integer(10) NOT NULL,
    FKpunktPośredni  integer(10),
    FKpasmoGórskie integer(10) NOT NULL,
    status       boolean NOT NULL,
    FOREIGN KEY(FKpunktPoczątkowy) REFERENCES PunktyOficjalne(id),
    FOREIGN KEY(FKpunktKońcowy) REFERENCES PunktyOficjalne(id),
    FOREIGN KEY(FKpunktPośredni) REFERENCES PunktyOficjalne(id),
    FOREIGN KEY(FKpasmoGórskie) REFERENCES PasmaGórskie(id));
```

```
CREATE TABLE DowodyOdcinków (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKodcinek integer(10) NOT NULL,
    FKdowód    integer(10) NOT NULL,
    FOREIGN KEY(FKodcinek) REFERENCES OdcinkiTras(id),
    FOREIGN KEY(FKdowód) REFERENCES Dowody(id));
```

```
CREATE TABLE Dowody (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    zdjęcie     blob,
    FKprzodownik integer(10),
    FOREIGN KEY(FKprzodownik) REFERENCES Przodownicy(nrLegitymacji),
    CONSTRAINT zdjęcieXORprzodownik
        CHECK ((Dowody.zdjęcie IS NULL OR Dowody.FKprzodownik IS NULL) AND
(Dowody.zdjęcie IS NOT NULL OR Dowody.FKprzodownik IS NOT NULL)));
```

```
CREATE TABLE PunktyOficjalne (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    nazwa       text,
    szerokośćGeo real(10) NOT NULL,
    długośćGeo   real(10) NOT NULL,
    FKpasmoGórskie integer(10) NOT NULL,
    FOREIGN KEY(FKpasmoGórskie) REFERENCES PasmaGórskie(id));
```

```
CREATE TABLE PunktyWłasne (
```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
FKturysta   integer(10) NOT NULL,
nazwa       text,
szerokośćGeo real(10) NOT NULL,
długośćGeo  real(10) NOT NULL,
FKpasmaGórskie integer(10) NOT NULL,
CONSTRAINT turysta_nazwa
    UNIQUE (FKturysta, nazwa),
FOREIGN KEY(FKturysta) REFERENCES Turyści(nrKsiążeczk),
FOREIGN KEY(FKpasmaGórskie) REFERENCES PasmaGórskie(id);

CREATE TABLE Odznaki (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKturysta   integer(10) NOT NULL,
    wymaganePunkty integer(10) NOT NULL,
    dataPrzynania text NOT NULL,
    rodzaj        RodzajOdznaki NOT NULL CREATE TYPE RodzajOdznaki AS
ENUM('popularna','mała'),
    stopień       StopieńOdznaki CREATE TYPE StopieńOdznaki AS
ENUM('brązowy','srebrny','złoty'),
    FOREIGN KEY(FKturysta) REFERENCES Turyści(nrKsiążeczk),
    CONSTRAINT popularna_bruk_stopnia
        CHECK ((Odznaki.rodzaj = 'popularna' AND Odznaki.stopień IS NULL) OR
Odznaki.rodzaj = 'mała'));

CREATE TABLE PasmaGórskie (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    nazwa       text NOT NULL UNIQUE,
    obszar      blob NOT NULL,
    FKgrupaGórskie integer(10) NOT NULL,
    FOREIGN KEY(FKgrupaGórskie) REFERENCES GrupyGórskie(id));

CREATE TABLE GrupyGórskie (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    nazwa       text NOT NULL UNIQUE,
    kraj        Kraj DEFAULT 'Polska' NOT NULL CREATE TYPE Kraj AS
ENUM('Polska','Czechy','Słowacja','Niemcy','Ukraina'),
    obszar      blob NOT NULL;

CREATE TABLE UprawnieniaPrzodowników (
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    FKprzodownik integer(10) NOT NULL,
    FKgrupaGórskie integer(10) NOT NULL,
    FOREIGN KEY(FKgrupaGórskie) REFERENCES GrupyGórskie(id),
    FOREIGN KEY(FKprzodownik) REFERENCES Przodownicy(nrLegitymacji));

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Wyniki etapu 4: Implementacja, testy jednostkowe, ocena jakości

E - książeczka GOT PTTK

Projektowanie oprogramowania



E - GOT PTTK

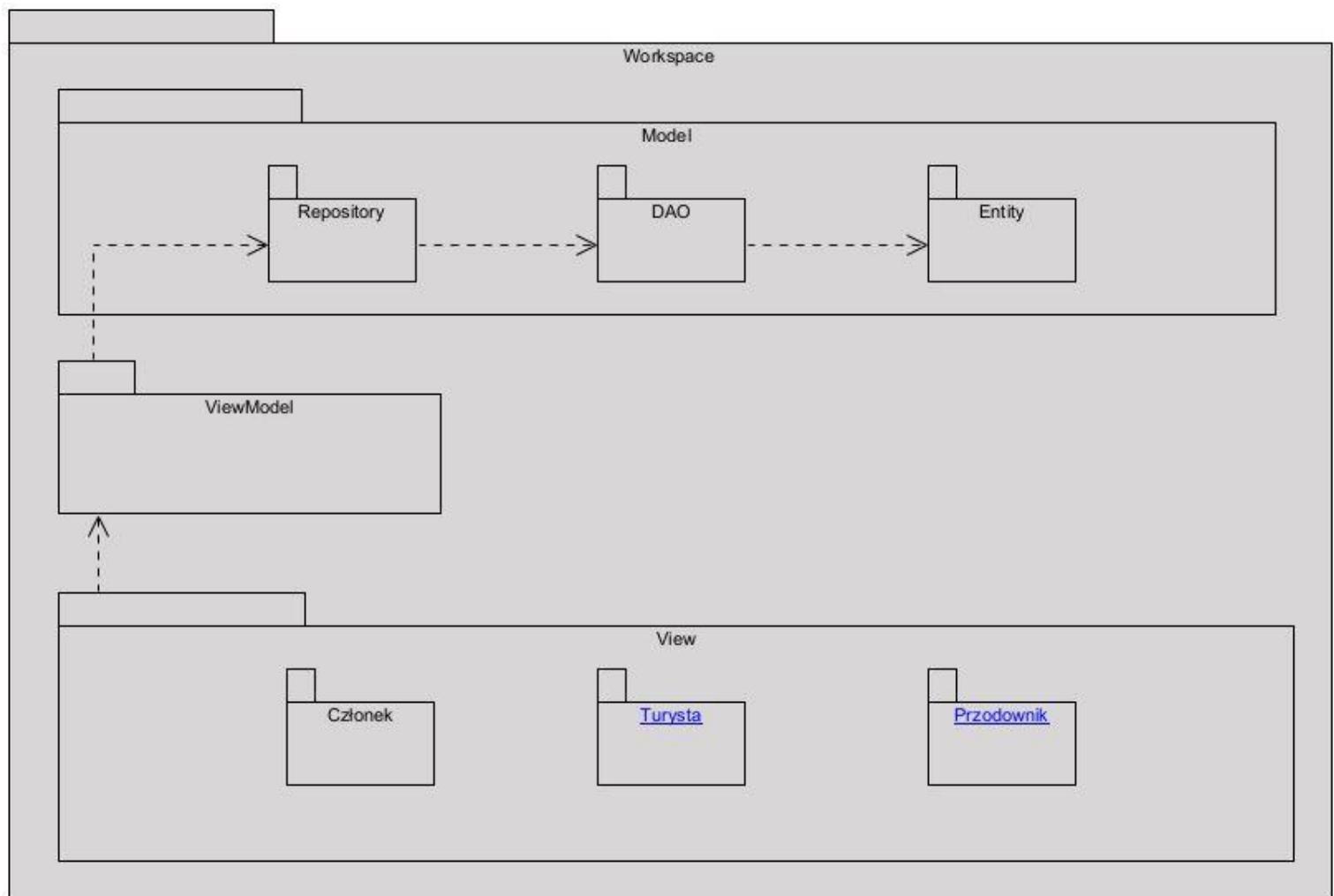
Skład zespołu:
Agnieszka Kłobus
Aleksandra Stecka
(zespół 8)

Prowadzący:
dr inż. Zbigniew Szpunar

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Zmiany do Etapu 3:

- zmieniono nazwę pakietu Pracownik na Członek (punkt 1 Etapu 3), ponieważ to Członek KTG ZG PTTK jest aktorem dla przypadku użycia Edycja odcinka w spisie.



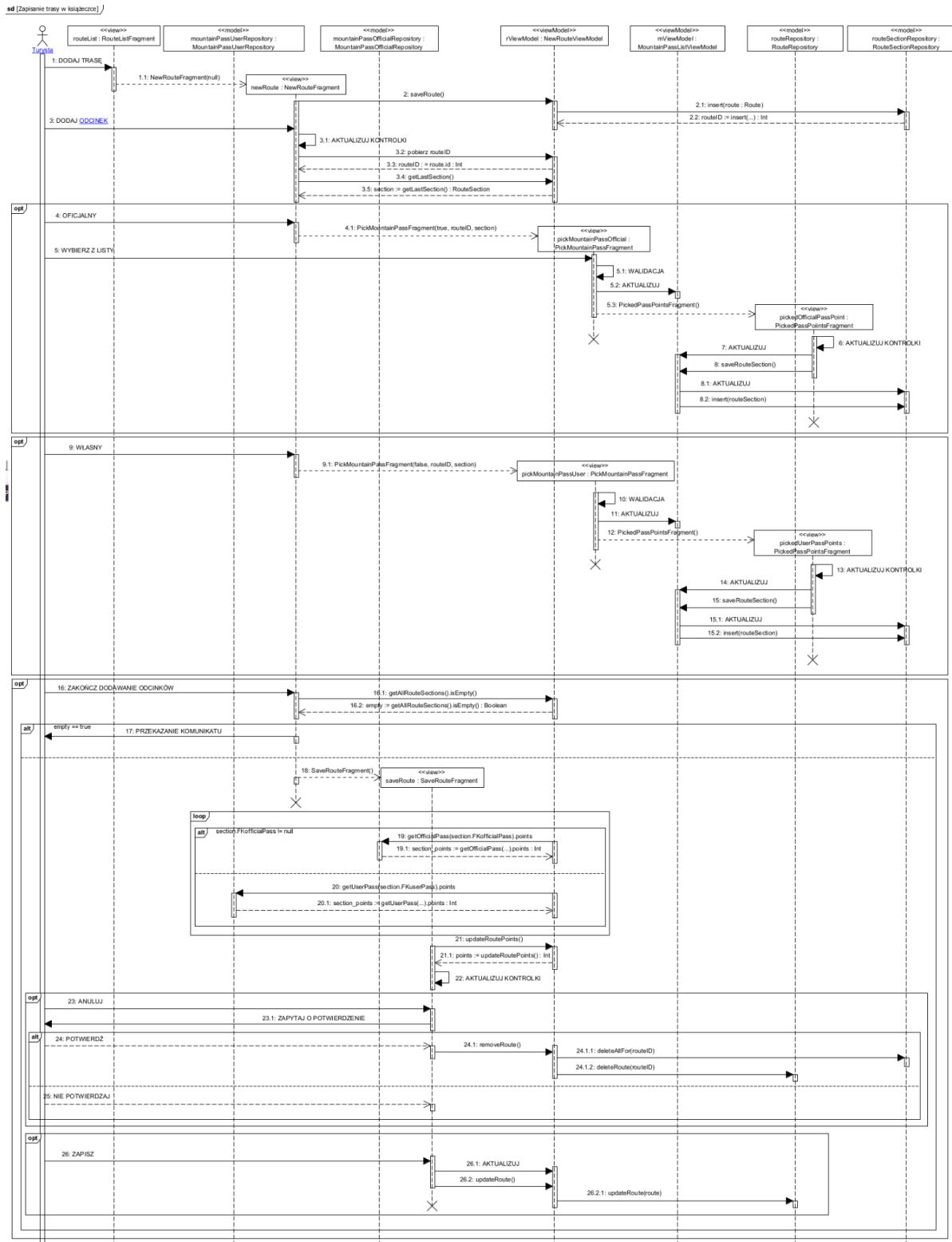
Uzupełnienie Etapu 3

Diagramy sekwencji zaczynają się od następnej strony.

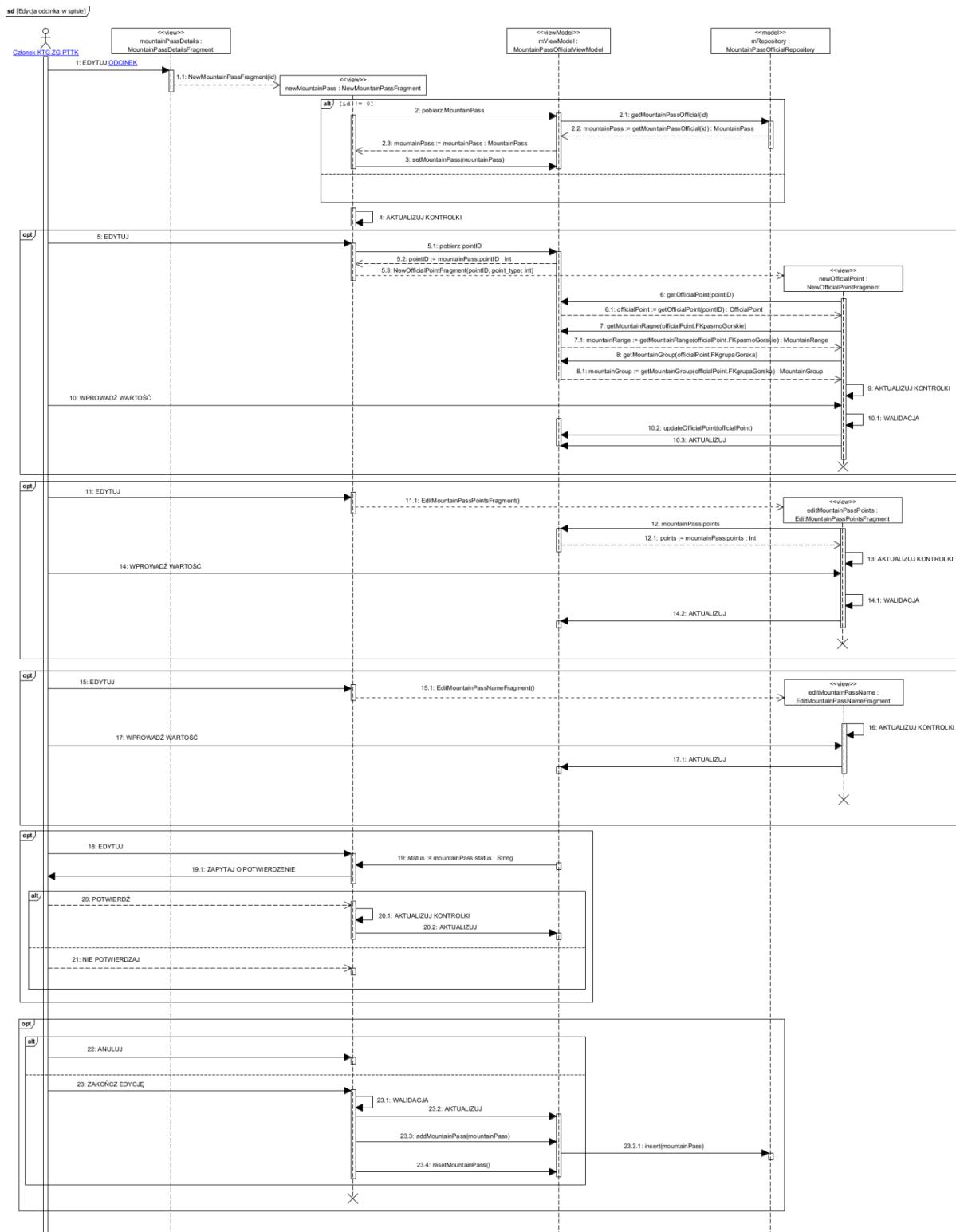
1. Zamieszczenie dowodu – Agnieszka Kłobus



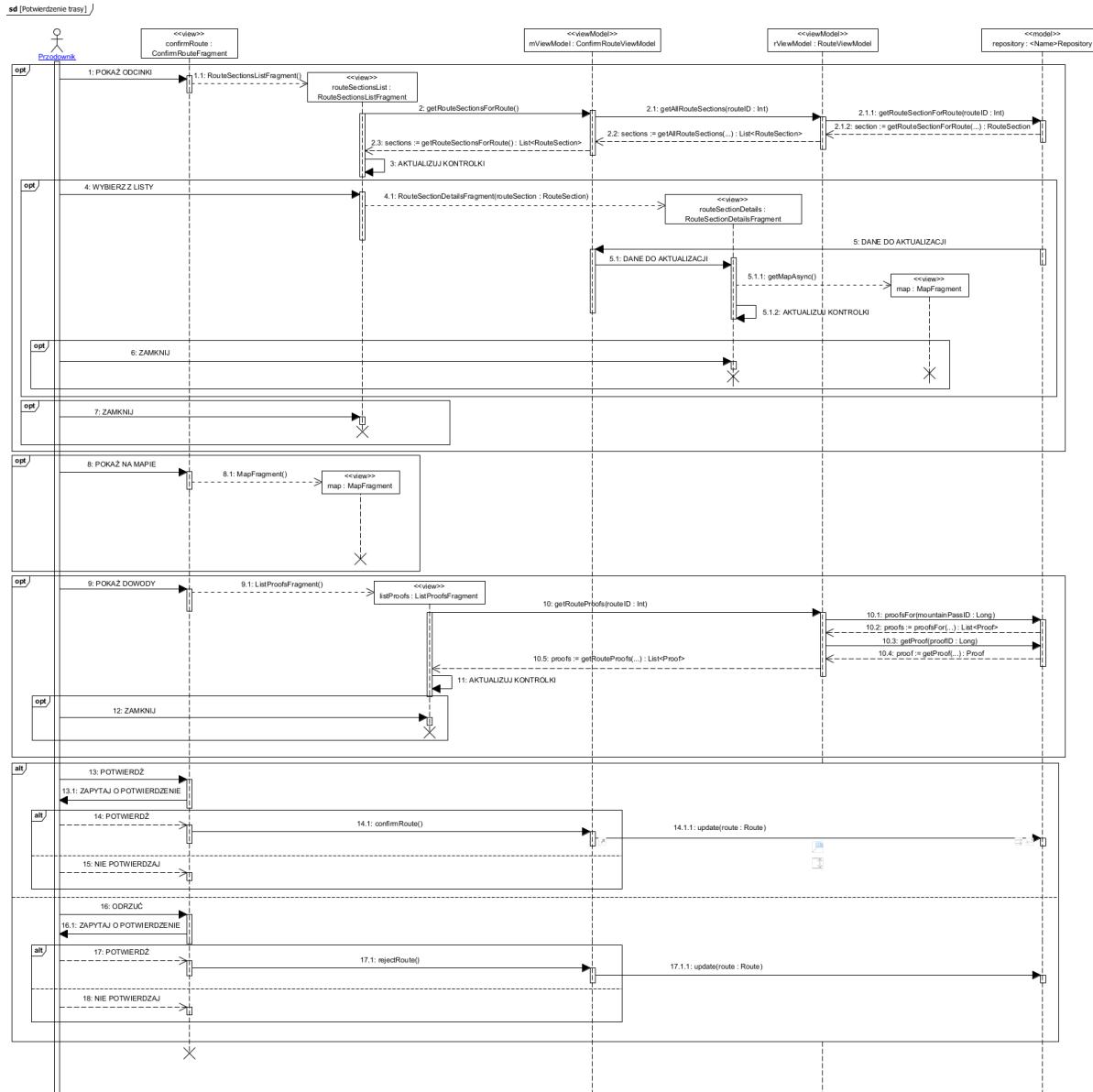
2. Zapisanie trasy w książeczkę – Agnieszka Kłobus



3. Edycja odcinka w spisie – Aleksandra Stecka



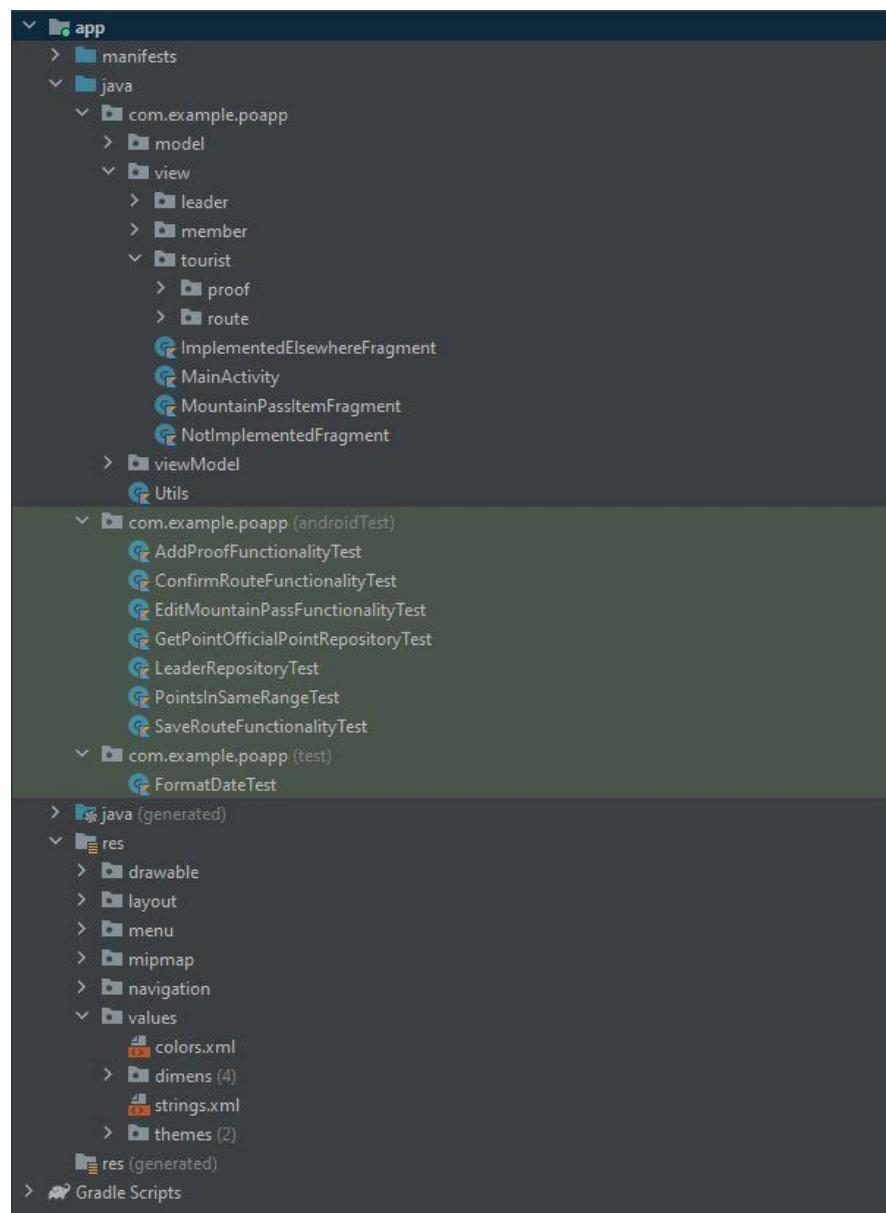
4. Potwierdzenie trasy – Aleksandra Stecka



E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Implementacja

5. Struktura kodu



Pakiety **dao**, **entity**, **repository**, a także **member**, **leader**, **tourist**, **proof** oraz **route** zostały wprowadzone w celu lepszej organizacji kodu.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

6. Wzorce projektowe

- Twórczy wzorzec projektowy Builder
 - Do utworzenia obiektu klasy nie jest wykorzystywany konstruktor. W przypadku użycia konstruktora wymagane jest podanie wartości dla wszystkich pól klasy, a czasem nie ma takiej potrzeby. W tej sytuacji wykorzystywana jest wewnętrzna klasa Builder, za której pomocą można utworzyć instancję klasy i wywołać tylko kilka wymaganych metod.
 - Przykładem zastosowania wzorca projektowego w projekcie jest tworzenie okien dialogowych do komunikacji z użytkownikiem – AlertDialog. Dzięki użyciu wzorca można wywołać tylko istotne w danym kontekście metody. W ten sposób utworzono okna dialogowe z jednym przyciskiem i z dwoma przyciskami.

```
val alertDialog = requireActivity().let { it: FragmentActivity
    val builder = AlertDialog.Builder(it)
    builder.apply { this: AlertDialog.Builder
        setNeutralButton("OK") { dialog, _ ->
            dialog.dismiss()
        }
        setTitle("Uwaga")
        setMessage("Dodaj przynajmniej jeden odcinek")
    }
    builder.create() ^let
}
alertDialog.show()
```

Wykorzystanie wzorca w kodzie – utworzenie okna dialogowego z jednym przyciskiem – metoda setNeutralButton(...)

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

val alertDialog = requireActivity().let { it: FragmentActivity
    val builder = AlertDialog.Builder(it)
    builder.apply { this: AlertDialog.Builder
        setPositiveButton("OK") { dialog, _ ->
            dialog.dismiss()
            mViewModel.resetMountainPass()
            activity?.supportFragmentManager?.popBackStack()
        }
        setNegativeButton("Cofnij") { dialog, _ ->
            dialog.dismiss()
        }
        setTitle("Uwaga")
        setMessage("Czy na pewno chcesz anulować?")
    }
    builder.create() ^let
}
alertDialog.show()

```

Wykorzystanie wzorca w kodzie – utworzenie okna dialogowego z dwoma przyciskami – metody setPositiveButton(...) i setNegativeButton(...)

- Twórczy wzorzec projektowy wstrzykiwania zależności
 - Dla obiektów klas zależnych od innych klas (jednym z pól klasy jest obiekt innej klasy) nie należy tworzyć obiektów klas wewnątrz klasy zależnej. Nie jest wtedy możliwe utworzenie zależności kilku obiektów od jednego obiektu (każdy obiekt klasy zależnej posiada własną instancję klasy, od której zależy).
 - Przykładem zastosowania wzorca projektowego w projekcie jest zależność obiektów MountainPassAdapter od obiektów MountainPassOfficialViewModel. Obiekt ViewModel nie jest tworzony wewnątrz klasy MountainPass Adapter, tylko przekazywany jako argument konstruktora.

```

class MountainPassAdapter(
    private val context: Context,
    private val values: List<MountainPassOfficial>,
    private val mViewModel: MountainPassOfficialViewModel,
    private val onMountainPassClickedListener: OnMountainPassClickedListener
) :

```

Wykorzystanie wzorca w kodzie – konstruktor klasy MountainPassAdapter

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```
list.adapter = MountainPassAdapter(activity as Context, allPasses, mViewModel, object : OnMountainPassClickedListener {
    override fun onItemClick(item: MountainPassOfficial) {
        activity?.supportFragmentManager?.beginTransaction()
            ?.replace(
                R.id.nav_host_fragment_activity_mountain_passes_list,
                MountainPassDetailsFragment(item.id)
            )
            ?.addToBackStack(null)
            ?.commit()
    }
})
```

Wykorzystanie wzorca w kodzie – tworzenie obiektu MountainPassAdapter

- Behawioralny wzorzec projektowy obserwator
 - Pewne obiekty powinny być zależne od zmian w innym obiekcie. Te zmiany to może być zmiana wartości atrybutów obiektu lub wystąpienie jakiegoś zdarzenia. Obserwatorzy / słuchacze mają być powiadamiani o zmianie stanu obiektu.
 - Przykładem zastosowania wzorca projektowego w projekcie są słuchacze zdarzeń, np kliknięcia przycisku. Słuchacz powiadamiany jest o zmianie stanu obiektu, którego jest słuchaczem. Zmiana stanu obiektu występuje w wyniku zdarzenia, najczęściej pochodzącego od użytkownika.

```
override fun onItemClick(item: MountainPassOfficial) {
    activity?.supportFragmentManager?.beginTransaction()
        ?.replace(
            R.id.nav_host_fragment_activity_mountain_passes_list,
            MountainPassDetailsFragment(item.id)
        )
        ?.addToBackStack( name: null)
        ?.commit()
}
```

Wykorzystanie wzorca w kodzie – słuchacz nasłuchuje wyboru elementu listy

- Wzorzec projektowy DAO
 - Pozwala na uniezależnienie aplikacji od źródła danych. Aplikacja wykorzystuje obiekty DAO w celu wykonania operacji na źródle danych, ale te obiekty DAO bezpośrednio komunikują się ze źródłem danych.
 - Wzorzec projektowy został zastosowany w projekcie w celu uniezależnienia aplikacji od źródła danych. To obiekty DAO wiedzą, jak

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

powinny wykonać operacje na danych – w tym przypadku wykorzystują język SQL.

```
@Dao
interface UserDAO {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insert(user: User): Long

    @Update(onConflict = OnConflictStrategy.REPLACE)
    fun update(user: User)

    @Query("select * from Uzytkownicy where id = :userId")
    fun getUser(userId: Long): List<User>

    @Delete
    fun delete(user: User)

    @Query("delete from Uzytkownicy")
    fun deleteAll()

    @Query("select * from Uzytkownicy")
    fun getAll(): LiveData<List<User>>
}
```

Wykorzystanie wzorca w kodzie – obiekt DAO operujący na tabeli użytkowników w bazie danych

- Wzorzec projektowy Repository
 - Jest to dodatkowa warstwa abstrakcji położona nad warstwą dostępu do danych – korzysta z obiektów DAO. Pozwala wykonywać operacje na kolekcjach obiektów.
 - Wzorzec projektowy został zastosowany w projekcie w celu uzyskania możliwości prostego pozyskiwania kolekcji danych ze źródła danych. Repozytoria wykorzystywane są głównie w klasach pakietu ViewModel.

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```
class UserRepository(private val userDao: UserDAO) {

    fun insert(user: User): Long {
        return userDao.insert(user)
    }

    fun update(user: User) {
        userDao.update(user)
    }

    fun delete(user: User) {
        userDao.delete(user)
    }

    fun deleteAll() {
        userDao.deleteAll()
    }

    fun getAll(): LiveData<List<User>> {
        return userDao.getAll()
    }

    fun getUser(id: Long): List<User> {
        return userDao.getUser(id)
    }
}
```

Wykorzystanie wzorca w kodzie – repozytorium operujące na tabeli użytkowników w bazie danych za pośrednictwem obiektu UserDAO

7. Repozytorium na serwisie Github

<https://github.com/AleksandraStecka/PO-21-22>

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Testowanie

8. Testy jednostkowe – Agnieszka Kłobus

8.1. test repozytorium OfficialPointRepository

Testy dotyczą metod

- `getOfficialPoint(Int)`
- `getOfficialPoint(String)`
- `insert(OfficialPoint)`

```
class GetPointOfficialPointRepositoryTest {

    private lateinit var db: AppDatabase
    private lateinit var officialPointRepository: OfficialPointRepository
    private lateinit var testPoint: OfficialPoint
    private var testPointID: Long? = null
    private var rangeID: Long? = null

    @Before
    fun createDB() {
        val applicationContext = ApplicationProvider.getApplicationContext<Context>()
        db = Room.inMemoryDatabaseBuilder(applicationContext,
AppDatabase::class.java).build()
        officialPointRepository =
OfficialPointRepository(db.officialPointDAO())
        val groupID = db.mountainGroupDAO().insert(MountainGroup(0, "Tatry i
Podtatrze", "Polska", byteArrayOf()))
        rangeID = db.mountainRangeDAO().insert(MountainRange(0, "Tatry
Wysokie", byteArrayOf(), groupID))
    }

    @After
    @Throws(IOException::class)
    fun closeDb() {
        db.close()
    }

    @Test
    fun getNullPoint() {
        val testPointIDs = officialPointRepository.getOfficialPoint(10)
        assertTrue(testPointIDs.isEmpty())
    }

    @Test
    fun insertPoint() {
        testPoint = OfficialPoint(0, "Rusinowa Polana", 49.261037737025546,
20.08954157827796, rangeID!!.toInt())
        testPointID = officialPointRepository.insert(testPoint)
    }
}
```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

        assertNotNull(testPointID)
    }

    @Test
    fun getResult() {
        insertPoint()
        var testPointIDs =
officialPointRepository.getOfficialPoint(testPointID!!.toInt())
        assertTrue(testPointIDs.isNotEmpty())
        assertTrue(testPointIDs.size == 1)
        testPointIDs =
officialPointRepository.getOfficialPoint(testPoint.nazwa)
        assertTrue(testPointIDs.isNotEmpty())
        assertTrue(testPointIDs.size == 1)
    }

    @Test
    fun getCorrectResultByID() {
        insertPoint()
        val testPointIDs =
officialPointRepository.getOfficialPoint(testPointID!!.toInt())
        assertEquals(testPointIDs[0].id, testPointID?.toInt())
        assertEquals(testPointIDs[0].nazwa, testPoint.nazwa)
        assertEquals(testPointIDs[0].FKpaskoGorskie, rangeID?.toInt())
    }

    @Test
    fun getCorrectResultByName() {
        insertPoint()
        val testPointIDs =
officialPointRepository.getOfficialPoint(testPoint.nazwa)
        assertEquals(testPointIDs[0].id, testPointID?.toInt())
        assertEquals(testPointIDs[0].nazwa, testPoint.nazwa)
        assertEquals(testPointIDs[0].FKpaskoGorskie, rangeID?.toInt())
    }
}

```

8.2. test metody pointsNotInTheSameRange w klasie Utils

```

class PointsInSameRangeTest {
    private var range1: Int? = null
    private var range2: Int? = null
    private var group: Int? = null
    private var group2: Int? = null
    private var point11: Int? = null
    private var point12: Int? = null
    private var point2: Int? = null

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

private lateinit var db: AppDatabase
private lateinit var repository: OfficialPointRepository

@DelicateCoroutinesApi
@Before
fun getTestPoints() {
    val applicationContext = ApplicationProvider.getApplicationContext<Context>()
    db = Room.inMemoryDatabaseBuilder(applicationContext,
AppDatabase::class.java).build()
    repository = OfficialPointRepository(db.officialPointDAO())
    group = db.mountainGroupDAO().insert(MountainGroup(0, "group", "", byteArrayOf())).toInt()
    group2 = db.mountainGroupDAO().insert(MountainGroup(0, "group2", "", byteArrayOf())).toInt()
    range1 = db.mountainRangeDAO().insert(MountainRange(0, "1", byteArrayOf(), group!!.toLong())).toInt()
    range2 = db.mountainRangeDAO().insert(MountainRange(0, "1", byteArrayOf(), group2!!.toLong())).toInt()
    point11 = db.officialPointDAO().insert(OfficialPoint(0, "1", 0.0, 0.0, range1!!)).toInt()
    point12 = db.officialPointDAO().insert(OfficialPoint(0, "2", 0.0, 0.0, range1!!)).toInt()
    point2 = db.officialPointDAO().insert(OfficialPoint(0, "3", 0.0, 0.0, range2!!)).toInt() }

    @After
    @Throws(IOException::class)
    fun deleteTestData() {
        db.close() }

    @Test
    fun startEndCorrect() {
        val result = Utils.pointsNotInSameRange(
            MountainPassOfficial(0, "", 1, point11!!, point12!!, null, 0, ""),
            repository
        )
        assertFalse(result) }

    @Test
    fun startEndIncorrect() {
        val result = Utils.pointsNotInSameRange(
            MountainPassOfficial(0, "", 1, point11!!, point2!!, null, 0, ""),
            repository
        )
        assertTrue(result) }

    @Test

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

fun endStartIncorrect() {
    val result = Utils.pointsNotInSameRange(
        MountainPassOfficial(0, "", 1, point2!!, point12!!, null, 0, ""),
        repository
    )
    assertTrue(result)
}

@Test
fun startEndCorrectThroughIncorrect() {
    val result = Utils.pointsNotInSameRange(
        MountainPassOfficial(0, "", 1, point11!!, point12!!, point2!!, 0, ""),
        repository
    )
    assertTrue(result)
}

@Test
fun startEndCorrectThroughCorrect() {
    val result = Utils.pointsNotInSameRange(
        MountainPassOfficial(0, "", 1, point11!!, point12!!, point12!!, 0, ""),
        repository
    )
    assertFalse(result)
}

@Test
fun startThroughCorrectEndIncorrect() {
    val result = Utils.pointsNotInSameRange(
        MountainPassOfficial(0, "", 1, point11!!, point2!!, point12!!, 0, ""),
        repository
    )
    assertTrue(result)
}
}

```

8. Testy jednostkowe – Aleksandra Stecka

8.1. test repozytorium LeaderRepository

Testy dotyczą metod

- `getLeader(Long)`
- `getLeaderUser(Long)`

```

@RunWith(AndroidJUnit4::class)
class GetLeaderLeaderRepositoryTest {
    private lateinit var leaderDAO: LeaderDAO

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

private lateinit var userDAO: UserDAO
private lateinit var touristDAO: TouristDAO
private lateinit var leaderRepository: LeaderRepository
private lateinit var db: AppDatabase
private var leaderID: Int? = null
private var userID: Int? = null

@Before
fun createDB() {
    val appContext = ApplicationProvider.getApplicationContext<Context>()
    db = Room.inMemoryDatabaseBuilder(appContext,
AppDatabase::class.java).build()
    leaderDAO = db.leaderDAO()
    userDAO = db.userDAO()
    touristDAO = db.touristDAO()
    leaderRepository = LeaderRepository(leaderDAO) }

@After
@Throws(IOException::class)
fun closeDb() {
    db.close() }

@Test
fun getLeaderNull() {
    val leaderNull = leaderRepository.getLeader(10)
    assertNull(leaderNull) }

@Test
fun insertLeader() {
    userID = userDAO.insert(User(0, "jkowalski", "p@ssw0rd",
"jkowalski@gmail.com", "Jan", "Kowalski", "27.02.1995", 1)).toInt()
    val touristID = touristDAO.insert(Tourist(0, userID!!.toInt(), 0,
false))
    leaderID = leaderDAO.insert(Leader(0, touristID.toInt())).toInt()
    assertNotNull(leaderID) }

@Test
fun findLeader() {
    insertLeader()
    val found = leaderID?.let { leaderRepository.getLeader(it.toLong()) }
    assertEquals(found?.nrLegitymacji, leaderID) }

@Test
fun findLeaderTourist() {
    insertLeader()
    val found = leaderID?.let {
leaderRepository.getLeaderUser(it.toLong()) }
    assertEquals(found?.id, userID)
}

```

E-ksiązeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

        assertEquals(found?.imie, "Jan")
        assertEquals(found?.nazwisko, "Kowalski")
        assertEquals(found?.rola, 1)
    }
}

```

8.2. test metody formatDate w klasie Utils

```

class FormatDateTest {

    @Test
    fun fullNumbers() {
        val result = Utils.formatDate(2000, 9, 10)
        assertEquals(result, "2000-10-10")
    }

    @Test
    fun singleMonth() {
        val result = Utils.formatDate(2000, 8, 10)
        assertEquals(result, "2000-09-10")
    }

    @Test
    fun singleDay() {
        val result = Utils.formatDate(2000, 9, 1)
        assertEquals(result, "2000-10-01")
    }

    @Test
    fun bothSingle() {
        val result = Utils.formatDate(2020, 8, 1)
        assertEquals(result, "2020-09-01")
    }

    @Test
    fun leapYearFebruary() {
        val result = Utils.formatDate(2020, 1, 29)
        assertEquals(result, "2020-02-29")
    }

    @Test(expected = IllegalArgumentException::class)
    fun notLeapYearFebruary() {
        Utils.formatDate(2021, 1, 29)
    }

    @Test(expected = IllegalArgumentException::class)
    fun incorrectDay() {
        Utils.formatDate(2021, 1, 30)
    }

    @Test(expected = IllegalArgumentException::class)
    fun incorrectMonthTooLow() {
        Utils.formatDate(2021, -2, 30)
    }

    @Test(expected = IllegalArgumentException::class)
    fun incorrectDayTooLow() {
        Utils.formatDate(2021, 1, 31)
    }
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

    Utils.formatDate(2021, 1, -2) }

@Test(expected = IllegalArgumentException::class)
fun incorrectYearTooLow() {
    Utils.formatDate(1998, -2, 30)
}
}

```

9. Przypadki testowe dla PU

9.1. Edycja odcinka

TC ID	Idea testowa	Kroki testu	Dane testowe	Oczekiwane wyniki
TC1	Sprawdzenie poprawności przejścia do edycji odcinka, wyświetlenie ekranu szczegółów odcinka	1. Kliknij przycisk "Edycja odcinka A.Stecka" 2. Kliknij pierwszy element listy	-	Wyświetlenie ekranu szczegółów odcinka Wyświetlenie ekranu edycji odcinka
TC2	Sprawdzenie poprawności edycji punktów odcinka	1. Przejdź do ekranu edycji pierwszego odcinka listy 2. Kliknij przycisk "Edytuj" dla punktu początkowego 3. Zmień nazwę odcinka 4. Kliknij przycisk "Zapisz" 5. Kliknij przycisk "Zakończ"	Zmieniona Nazwa	Wyświetlenie nowej nazwy punktu początkowego odcinka w szczegółach oraz liście
TC3	Sprawdzenie poprawności walidacji pasma górskiego przy edycji punktu	1. Przejdź do ekranu edycji pierwszego odcinka listy 2. Kliknij przycisk "Edytuj" dla punktu pośredniego 3. Zmień pasmo górskie 4. Kliknij przycisk "Zapisz"	Nowe Pasmo Test	Wyświetlenie okna dialogowego informującego o podaniu złego pasma górskiego
TC4	Sprawdzenie poprawności walidacji liczby punktów przy edycji odcinka i edycji tej liczby	1. Przejdź do ekranu edycji pierwszego odcinka listy 2. Kliknij przycisk "Edytuj" dla ilości punktów 3. Zmień ilość punktów 4. Kliknij przycisk "Zapisz" 5. Zamknij okno dialogowe 6. Zmień ilość punktów 7. Kliknij przycisk "Zapisz" 8. Kliknij przycisk "Zakończ"	-5 10	Wyświetlenie okna dialogowego dla punktów ujemnych oraz zapisane i wyświetlenie zmienionej liczby punktów dla poprawnej wartości
TC5	Sprawdzenie działania zmiany statusu odcinka	1. Przejdź do ekranu edycji pierwszego odcinka listy 2. Kliknij przycisk "Edytuj" dla statusu 3. Zaakceptuj zmiany 4. Wróć do widoku listy		Wyświetlenie okna dialogowego potwierdzenia zmiany, zmiana statusu odcinka, wyświetlenie zmian w liście odcinków

9.2. Zapisanie trasy w książeczkę

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

TC ID	Idea testowa	Kroki testu	Dane testowe	Oczekiwane wyniki
TC6	Sprawdzenie poprawności wyświetlania ekranu tworzenia trasy i weryfikacji przechodzenia do dodawaniu dowodów oraz zapisana trasy bez dodania odcinków	1. Kliknij "Zapisanie trasy i zamieszczenie dowodu A. Kłobus" 2. Kliknij "Dodaj trasę" 3. Kliknij "Dodaj dowód" 4. Zamknij okno dialogowe 5. Kliknij "Zakończ dodawanie odcinków" 6. Zamknij okno dialogowe		Poprawne wyświetlenie layoutów Wyświetlenie okna dialogowego przy próbie dodania dowodu gdy nie ma odcinków, Wyświetlenie okna dialogowego przy próbie zapisania pustej tarsy
TC7	Sprawdzenie poprawności dodawania odcinka do trasy oraz weryfikacji czasu	1. Przejście do dodawania nowej trasy 2. Kliknij "Dodaj odcinek" 3. Kliknij "Wybierz własny" 4. Wybranie pierwszego danego odcinka 5. Podanie czasu -5 6. Zamknięcie okna dialogowego 7. Podanie czasu 20 8. Kliknięcie "Dodaj odcinek"	Odcinek Własny 1 – Od Parking TPN do Rusinowa Polana Punkty – 6 Czas – -5 Czas – 20	Poprawne wyświetlenie listy odcinków, Wybranie odcinka z listy i przejście do widoku wpisania czasu, Walidacja i wyświetlenie komunikatu przy podaniu ujemnej wartości, Wyświetlenie odcinka w liście po podaniu prawidłowego czasu
TC8	Sprawdzenie poprawności tworzenia trasy z kilkoma odcinkami różnego typu	1. Przejście do dodawania nowej trasy 2. Kliknij "Dodaj odcinek" 3. Kliknij "Wybierz własny" 4. Wybranie pierwszego danego odcinka 5. Podanie czasu 1 6. Kliknij "Dodaj odcinek" 7. Kliknij "Dodaj odcinek" 8. Kliknij "Wybierz ze spisu" 9. Wybranie odcinka 2 10. Podanie czasu 2 11. Kliknij "Dodaj odcinek" 12. Kliknij "Dodaj odcinek" 13. Kliknij "Wybierz ze spisu" 14. Wybranie odcinka 3 15. Podanie czasu 3 16. Kliknij "Dodaj odcinek" 17. Kliknij "Zakończ dodawanie odcinków" 18. Kliknij "Zapisz trasę"	Odcinek Własny 1 – Od Parking TPN do Rusinowa Polana Punkty – 6 Czas 1 – 50 Odcinek 2 – Od Rusinowa Polana do Wierch poroniec Punkty – 4 Czas 2 – 30 Odcinek 3 – Od Wierch Poroniec do Łysa Polana Punkty – 3 Czas 3 – 80 Data – Today	Poprawne dodanie odcinków, list odcinków, wyświetlanie listy odcinków trasy
TC9	Sprawdzenie poprawności weryfikacji ciągłości trasy	1. Przejście do dodawania nowej trasy 2. Kliknij "Dodaj odcinek" 3. Kliknij "Wybierz własny" 4. Wybranie pierwszego danego odcinka 5. Podanie czasu 1 6. Kliknij "Dodaj odcinek" 7. Kliknij "Dodaj odcinek" 8. Kliknij "Wybierz ze spisu" 9. Wybranie odcinka 2	Odcinek Własny 1 – Od Parking TPN do Rusinowa Polana Punkty – 6 Czas 1 – 50 Odcinek 2 – Od Wierch Poroniec do Łysa Polana Punkty – 3	Wyświetlenie komunikatu o potrzebie zgodności w punktach trasy

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

9.3. Dodanie dowodu

TC ID	Idea testowa	Kroki testu	Dane testowe	Oczekiwane wyniki
TC10	Sprawdzenie poprawności wyświetlania ekranów dodawania dowodu przy tworzeniu nowej trasy	<ol style="list-style-type: none"> 1. Kliknij "Zapisanie trasy i zamieszczenie dowodu A. Kłobus" 2. Kliknij "Dodaj trasę" 3. Kliknij "Dodaj dowód" 4. Zamknij okno dialogowe 5. Dodaj odcinek testowy 6. Kliknij "Dodaj dowód" 7. Kliknij "Dodaj zdjęcie" 8. Wróć 9. Kliknij "Dodaj dane przodownika" 	Odcinek Własny 1 – Od Parking TPN do Rusinowa Polana Punkty – 6 Czas 1 – 50	Poprawne wyświetlanie wszystkich przycisków i layoutów
TC11	Sprawdzenie poprawności wyświetlania ekranów przy edycji trasy	<ol style="list-style-type: none"> 1. Kliknij "Zapisanie trasy i zamieszczenie dowodu A. Kłobus" 2. Dodaj przykładową trasę 3. Kliknij w przykładową trasę 4. Kliknij "Edytuj dowody" 5. Kliknij "Dodaj dowód" 6. Kliknij "Dodaj zdjęcie" 7. Wróć 8. Kliknij "Dodaj dane przodownika" 9. Wróć 10. Wróć 11. Kliknij "Zobacz dowody" 12. Zamknij okno dialogowe 13. Kliknij "Usuń dowód" 14. Zamknij okno dialogowe 	Trasa testowa (Odcinek Własny 1 – Od Parking TPN do Rusinowa Polana Punkty – 6 Czas 1 – 50 Data – Today)	Poprawne wyświetlanie wszystkich przycisków i layoutów, przechodzenie między layoutami
TC12	Sprawdzenie poprawności dodania przodownika jako dowodu, wyświetlania oraz usuwania	<ol style="list-style-type: none"> 1. Utwórz testową trasę 2. Kliknij w przykładową trasę 3. Kliknij "Edytuj dowody" 4. Kliknij "Dodaj dowód" 5. Kliknij "Dodaj dane przodownika" 6. Wprowadź id przodownika 7. Kliknij "Zapisz" 8. Kliknij "Zobacz dowody" 9. Kliknij "Zamknij" 10. Kliknij "Usuń dowód" 11. Kliknij element listy 12. Kliknij "Usuń" 13. Zaakceptuj okno dialogowe 14. Kliknij zobacz dowody 15. Zamknij okno dialogowe 	Trasa testowa (Odcinek Własny 1 – Od Parking TPN do Rusinowa Polana Punkty – 6 Czas 1 – 50 Odcinek 2 – Od Rusinowa Polana do Wierch poroniec Punkty – 4 Czas 2 – 30 Odcinek 3 – Od Wierch Poroniec do Łysa Polana Punkty – 3 Czas 3 – 80 Data – Today) Id przodownika – 1	Poprawne dodanie danych, Wyświetlenie danych przodownika jako dowodu do wszystkich odcinków tras, Poprawne usunięcie przodownika i wyświetlenie informacji o braku dowodów
TC13	Sprawdzenie poprawności wybierania odcinków do	<ol style="list-style-type: none"> 1. Utwórz testową trasę 2. Kliknij "Edytuj dowody" 	Trasa testowa (Odcinek Własny 1 –	Poprawne zaznaczanie

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

	dowodu w postaci obrazu	<p>3. Kliknij "Dodaj dowód"</p> <p>4. Kliknij pierwszy element listy</p> <p>5. Kliknij drugi element listy</p> <p>6. Kliknij wybierz obraz</p>	<p>Od Parking TPN do Rusinowa Polana Punkty – 6 Czas 1 – 50 Odcinek 2 – Od Rusinowa Polana do Wierch poroniec Punkty – 4 Czas 2– 30 Odcinek 3 – Od Wierch Poroniec do Łysa Polana Punkty – 3 Czas 3 – 80 Data – Today)</p>	fragmentów trasy do dodania dowodu, rozpoczęcie działania aplikacji zewnętrznej w celu dodania obrazu
TC14	Sprawdzenie poprawności informowania o istniejących dowodach	<p>10. Kliknij "Zapisanie trasy i zamieszczenie dowodu A. Kłobus"</p> <p>11. Kliknij "Dodaj trasę"</p> <p>12. Dodaj odcinki testowe</p> <p>13. Kliknij "Dodaj dowód"</p> <p>14. Kliknij "Dodaj dane przodownika"</p> <p>15. Wprowadź id przodownika</p> <p>16. Kliknij "Zapisz"</p>	<p>Odcinki testowe (Odcinek Własny 1 – Od Parking TPN do Rusinowa Polana Punkty – 6 Czas 1 – 50 Odcinek 2 – Od Rusinowa Polana do Wierch poroniec Punkty – 4 Czas 2– 30) Id przodownika – 1</p>	Poprawne wyświetlanie informacji o braku dowodu przy odcinku oraz o wystąpieniu dowodu dla odcinka

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

9.4. Potwierdzenie trasy

TC ID	Idea testowa	Kroki testu	Dane testowe	Oczekiwane wyniki
TC15	Sprawdzenie poprawności wyświetlania ekranów przy potwierdzaniu trasy	<ol style="list-style-type: none"> Utwórz przykładową trasę z zamieszczonym dowodem i wyślij do potwierdzenia Kliknij "Potwierdzenie trasy A. Stecka" Wybierz pierwszy element z listy Kliknij "Pokaż odcinki" Wybierz pierwszy element z listy Kliknij "Zamknij" Kliknij "Zamknij" Kliknij "Pokaż na mapie" Kliknij przycisk cofania Kliknij "Pokaż dowody" Kliknij "Zamknij" Kliknij "Odrzuć" Kliknij "Cofnij" Kliknij "Potwierdź" Kliknij "Cofnij" 		Poprawne wyświetlanie wszystkich przycisków, layoutów i okien dialogowych
TC16	Sprawdzenie poprawności zatwierdzania trasy	<ol style="list-style-type: none"> Utwórz przykładową trasę z zamieszczonym dowodem i wyślij do potwierdzenia Kliknij "Potwierdzenie trasy A. Stecka" Wybierz pierwszy element z listy Kliknij "Potwierdź" Kliknij "OK" Kliknij przycisk cofania Kliknij "Zapisanie trasy i zamieszczenie dowodu A. Kłobus" Wybierz pierwszy element z listy 		Poprawne zatwierdzanie tras, nie wyświetlanie zatwierdzonych tras u przodownika, zmiana statusu trasy u turysty na "zatwierdzona"
TC17	Sprawdzenie poprawności odrzucania trasy	<ol style="list-style-type: none"> Utwórz przykładową trasę z zamieszczonym dowodem i wyślij do potwierdzenia Kliknij "Potwierdzenie trasy A. Stecka" Wybierz pierwszy element z listy Kliknij "Odrzuć" Kliknij "OK" Kliknij przycisk cofania Kliknij "Zapisanie trasy i zamieszczenie dowodu A. Kłobus" Wybierz pierwszy element z listy 		Poprawne odrzucanie tras, nie wyświetlanie odrzuconych tras u przodownika, zmiana statusu trasy u turysty na "zatwierdzona"

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

10. Automatyzacja testów funkcjonalnych

Testy funkcjonalne zautomatyzowane zostały jako kolejne testy w projekcie, przy pomocy framework'u Espresso.

10.1. Edycja odcinka

```
class EditMountainPassFunctionalityTest {

    @Rule
    @JvmField
    val mActivityRule = ActivityTestRule(MainActivity::class.java, false,
true)

    @Test
    fun testShowLayout() {
        onView(withId(R.id.mountainPass)).check(matches(isDisplayed()))
        onView(withId(R.id.mountainPass)).perform(click())

        onView(withId(R.id.mountain_passes_list)).perform(RecyclerViewActions.actionO
nItemAtPosition<MountainPassAdapter.MountainPassOfficialItemHolder>(0,
click()))
        onView(withText("Szczegóły")).check(matches(isDisplayed()))
    }

    @Test
    fun testEditPoint() {
        val changedName = "Zmieniona nazwa"
        onView(withId(R.id.mountainPass)).check(matches(isDisplayed()))
        onView(withId(R.id.mountainPass)).perform(click())

        onView(withId(R.id.mountain_passes_list)).perform(RecyclerViewActions.actionO
nItemAtPosition<MountainPassAdapter.MountainPassOfficialItemHolder>(0,
click()))
        onView(withId(R.id.edit_button)).perform(click())
        onView(withId(R.id.edit_start)).perform(click())

        onView(withId(R.id.official_point_name)).perform(replaceText(changedName))
        onView(withId(R.id.save_official_point)).perform(click())
        onView(withId(R.id.new_start)).check(matches(withText(changedName)))
        onView(withId(R.id.save_mountain_pass)).perform(click())
        onView(withText(changedName)).check(matches(isDisplayed()))
    }

    @Test
    fun testPointMountainRangeValidation() {
        val changeRange = "Nowe Pasmo Test"
    }
}
```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

    val alert = "Uwaga"
    onView(withId(R.id.mountainPass)).check(matches(isDisplayed()))
    onView(withId(R.id.mountainPass)).perform(click())

    onView(withId(R.id.mountain_passes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<MountainPassAdapter.MountainPassOfficialItemHolder>(0,
    click()))
        onView(withId(R.id.edit_button)).perform(click())
        onView(withId(R.id.edit_through)).perform(click())
        onView(withId(R.id.mountain_range)).perform(replaceText(changeRange))
        onView(withId(R.id.save_official_point)).perform(click())
        onView(withText(alert)).check(matches(isDisplayed()))
        onView(withText("OK")).perform(click())
    }

@Test
fun testEditPassPoints() {
    val incorrectPoints = "-5"
    val correctPoints = "10"
    val alert = "Uwaga"
    onView(withId(R.id.mountainPass)).check(matches(isDisplayed()))
    onView(withId(R.id.mountainPass)).perform(click())

    onView(withId(R.id.mountain_passes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<MountainPassAdapter.MountainPassOfficialItemHolder>(0,
    click()))
        onView(withId(R.id.edit_button)).perform(click())
        onView(withId(R.id.edit_points)).perform(click())

    onView(withId(R.id.edit_mountain_pass_points)).perform(replaceText(incorrectPoints))
        onView(withId(R.id.save_mountain_pass_points)).perform(click())
        onView(withText(alert)).check(matches(isDisplayed()))
        onView(withText("OK")).perform(click())

    onView(withId(R.id.edit_mountain_pass_points)).perform(replaceText(correctPoints))
        onView(withId(R.id.save_mountain_pass_points)).perform(click())
        onView(withId(R.id.save_mountain_pass)).perform(click())
        onView(withText(correctPoints)).check(matches(isDisplayed()))
    }

@Test
fun testChangeStatus() {
    val alert = "Uwaga"
    val active = "aktywny"
    val inactive = "zlikwidowany"
    onView(withId(R.id.mountainPass)).check(matches(isDisplayed()))
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

onView(withId(R.id.mountainPass)).perform(click())

onView(withId(R.id.mountain_passes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<MountainPassAdapter.MountainPassOfficialItemHolder>(0,
click()))
    onView(withId(R.id.edit_button)).perform(click())
    val status = getText(withId(R.id.new_status))
    val result = if (status == active) {
        inactive
    } else {
        active
    }
    onView(withId(R.id.edit_status)).perform(click())
    onView(withText(alert)).check(matches(isDisplayed()))
    onView(withText("OK")).perform(click())
    onView(withText(result)).check(matches(isDisplayed()))
    onView(withId(R.id.save_mountain_pass)).perform(click())
    onView(withText(result)).check(matches(isDisplayed()))
    Espresso.pressBack()
    onView(withId(R.id.mountain_passes_list))
        .check(matches(hasDescendant(withText(result))))
}

private fun getText(matcher: Matcher<View?>?): String? {
    val stringHolder = arrayOf<String?>(null)
    onView(matcher).perform(object : ViewAction {
        override fun getConstraints(): Matcher<View> {
            return isAssignableFrom(TextView::class.java)
        }

        override fun getDescription(): String {
            return "getting text from a TextView"
        }

        override fun perform(uiController: UiController?, view: View) {
            val tv = view as TextView
            stringHolder[0] = tv.text.toString()
        }
    })
    return stringHolder[0]
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

Tests	Duration	po_app
✓ Test Results	28 s	5/5
✓ EditMountainPassFunctionalityTest	28 s	5/5
✓ testPointMountainRangeValidation	5 s	✓
✓ testEditPoint	6 s	✓
✓ testShowLayout	3 s	✓
✓ testEditPassPoints	5 s	✓
✓ testChangeStatus	6 s	✓

10.2. Zapisanie trasy w książeczce

```
class SaveRouteFunctionalityTest {
    private val alert = "Uwaga"

    @Rule
    @JvmField
    val mActivityRule = ActivityTestRule(MainActivity::class.java, false,
true)

    @Test
    fun testShowLayout() {
        onView(withId(R.id.saveRouteAddProof)).check(matches(isDisplayed()))
        onView(withId(R.id.saveRouteAddProof)).perform(click())
        onView(withId(R.id.add_route_button)).perform(click())
        onView(withId(R.id.add_proof_button)).perform(click())
        onView(withText(alert)).check(matches(isDisplayed()))
        onView(withText("OK")).perform(click())
        onView(withId(R.id.end_button)).perform(click())
        onView(withText(alert)).check(matches(isDisplayed()))
        onView(withText("OK")).perform(click())
    }

    @Test
    fun testAddSectionVerifyTime() {
        val point1 = "Parking TPN"
        val point2 = "Rusinowa Polana"
        val pass1points = "6"
        val time1 = "-5"
        val time2 = "20"
        onView(withId(R.id.saveRouteAddProof)).check(matches(isDisplayed()))
        onView(withId(R.id.saveRouteAddProof)).perform(click())
        onView(withId(R.id.add_route_button)).perform(click())
        onView(withId(R.id.add_mountain_pass_button)).perform(click())
        onView(withId(R.id.own_button)).perform(click())

        onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.sc

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

rollToPosition<MountainPassPickAdapter<MountainPassOfficial>.MountainPassItem
Holder>(5))
    onViewMatchers.allOf(withChild(withText(point1)),
withChild(withText(point2)),
withChild(withText(pass1points))).perform(click())
        onView(withId(R.id.time_value)).perform(replaceText(time1))
        onView(withId(R.id.add_route_pass_button)).perform(click())
        onView(withId(R.id.alert)).check(matches(isDisplayed()))
        onView(withId(R.id.time_value)).perform(replaceText(time2))
        onView(withId(R.id.add_route_pass_button)).perform(click())
    }
}

@LargeTest
@Test
fun testAddSectionAndTime() {
    val point1 = "Parking TPN"
    val point2 = "Rusinowa Polana"
    val pass1points = "6"
    val time1 = "50"
    val point3 = "Wierch Poroniec"
    val pass2points = "4"
    val time2 = "30"
    val point4 = "Łysa Polana"
    val pass3points = "3"
    val time3 = "20"
    onView(withId(R.id.saveRouteAddProof)).check(matches(isDisplayed()))
    onView(withId(R.id.saveRouteAddProof)).perform(click())
    onView(withId(R.id.add_route_button)).perform(click())
    onView(withId(R.id.add_mountain_pass_button)).perform(click())
    onView(withId(R.id.own_button)).check(matches(isDisplayed()))
    onView(withId(R.id.own_button)).perform(click())

    onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.sc
rollToPosition<MountainPassPickAdapter<MountainPassOfficial>.MountainPassItem
Holder>(5))
        onViewMatchers.allOf(withChild(withText(point1)),
withChild(withText(point2)),
withChild(withText(pass1points))).perform(click())
            onView(withId(R.id.time_value)).perform(replaceText(time1))
            onView(withId(R.id.add_route_pass_button)).perform(click())
            onView(withId(R.id.alert)).check(matches(isDisplayed()))
            onView(withId(R.id.time_value)).perform(replaceText(time2))
            onView(withId(R.id.add_mountain_pass_button)).perform(click())
            onView(withId(R.id.official_button)).perform(click())
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

        onView(Matchers.allOf(withChild(withText(point3)),
withChild(withText(point2)),
withChild(withText(pass2points)))).perform(click())
            onView(withText(point2)).check(matches(isDisplayed()))
            onView(withId(R.id.time_value)).perform(replaceText(time2))
            onView(withId(R.id.add_route_pass_button)).perform(click())
            onView(Matchers.allOf(withChild(withText(point3)),
withChild(withText(point2)),
withChild(withText(pass2points)))).check(matches(isDisplayed()))
                onView(withId(R.id.add_mountain_pass_button)).perform(click())
                onView(withId(R.id.official_button)).perform(click())

onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.scrollToPosition<MountainPassPickAdapter<MountainPassOfficial>.MountainPassItemHolder>(8))
            onView(Matchers.allOf(withChild(withText(point3)),
withChild(withText(point4)),
withChild(withText(pass3points)))).perform(click())
                onView(withText(point4)).check(matches(isDisplayed()))
                onView(withId(R.id.time_value)).perform(replaceText(time3))
                onView(withId(R.id.add_route_pass_button)).perform(click())
                onView(Matchers.allOf(withChild(withText(point3)),
withChild(withText(point4)),
withChild(withText(pass3points)))).check(matches(isDisplayed()))
                    onView(withId(R.id.end_button)).perform(click())
                    onView(withId(R.id.choose_date_label)).check(matches(isDisplayed()))
                    onView(withId(R.id.save_route_btn)).perform(click())
    }

@Test
fun testRoutePointsVerity() {
    val point1 = "Parking TPN"
    val point2 = "Rusinowa Polana"
    val pass1points = "6"
    val time1 = "50"
    val point3 = "Wierch Poroniec"
    val point4 = "Łysa Polana"
    val pass2points = "3"
    onView(withId(R.id.saveRouteAddProof)).check(matches(isDisplayed()))
    onView(withId(R.id.saveRouteAddProof)).perform(click())
    onView(withId(R.id.add_route_button)).perform(click())
    onView(withId(R.id.add_mountain_pass_button)).perform(click())
    onView(withId(R.id.own_button)).perform(click())

onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.scrollToPosition<MountainPassPickAdapter<MountainPassOfficial>.MountainPassItemHolder>(5))
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

        onViewMatchers.allOf(withId(withText(point1)),
withId(withId(point2)),
withId(withId(pass1points))).perform(click())
        onView(withId(R.id.time_value)).perform(replaceText(time1))
        onView(withId(R.id.add_route_pass_button)).perform(click())
        onView(withId(R.id.add_mountain_pass_button)).perform(click())
        onView(withId(R.id.official_button)).perform(click())

onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.scrollToPosition<MountainPassPickAdapter<MountainPassOfficial>.MountainPassItemHolder>(8))
        onView(withId(point3),
withId(withId(point4)),
withId(withId(pass2points))).perform(click())
        onView(withId(alert)).check(matches(isDisplayed()))
    }

}

```

Tests	Duration	po_app
Test Results	26 s	4/4
SaveRouteFunctionalityTest	26 s	4/4
testShowLayout	6 s	✓
testRoutePointsVerify	6 s	✓
testAddSectionVerifyTime	5 s	✓
testAddSectionAndTime	8 s	✓

10.3. Dodanie dowodu

```

class AddProofFunctionalityTest {
    private val alert = "Uwaga"

    @Rule
    @JvmField
    val mActivityRule = ActivityTestRule(MainActivity::class.java, false,
true)

    private fun addTestRouteSection() {
        val point1 = "Parking TPN"
        val point2 = "Rusinowa Polana"
        val pass1points = "6"
        val time1 = "50"
        onView(withId(R.id.add_mountain_pass_button)).perform(click())
        onView(withId(R.id.own_button)).perform(click())

onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.sc

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

rollToPosition<MountainPassPickAdapter<MountainPassOfficial>.MountainPassItem
Holder>(5))
    onView(Matchers.allOf(withChild(withText(point1)),
withChild(withText(point2)),
withChild(withText(pass1points)))).perform(click())
    onView(withText(point1)).check(matches(isDisplayed()))
    onView(withId(R.id.time_value)).perform(replaceText(time1))
    onView(withId(R.id.add_route_pass_button)).perform(click())}

private fun addTestRouteShort() {
    onView(withId(R.id.add_route_button)).perform(click())
    addTestRouteSection()
    onView(withId(R.id.end_button)).perform(click())
    onView(withId(R.id.choose_date_label)).check(matches(isDisplayed()))
    onView(withId(R.id.save_route_btn)).perform(click())}

@Test
fun testShowLayoutNewRoute() {
    onView(withId(R.id.saveRouteAddProof)).check(matches(isDisplayed()))
    onView(withId(R.id.saveRouteAddProof)).perform(click())
    onView(withId(R.id.add_route_button)).perform(click())
    onView(withId(R.id.add_proof_button)).check(matches(isDisplayed()))
    onView(withId(R.id.add_proof_button)).perform(click())
    onView(withText(alert)).check(matches(isDisplayed()))
    onView(withText("OK")).perform(click())
    addTestRouteSection()
    onView(withId(R.id.add_proof_button)).perform(click())
    onView(withId(R.id.add_image)).check(matches(isDisplayed()))
    onView(withId(R.id.add_leader)).check(matches(isDisplayed()))
    onView(withId(R.id.add_image)).perform(click())
    Espresso.pressBack()
    onView(withId(R.id.add_leader)).perform(click())}

@Test
fun testShowLayoutEditRoute() {
    onView(withId(R.id.saveRouteAddProof)).perform(click())
    addTestRouteShort()

    onView(withId(R.id.last_routes_list)).perform(RecyclerViewActions.actionOnIte
mAtPosition<RouteAdapter.RouteItemHolder>(0, click()))
    onView(withId(R.id.edit_proofs)).perform(click())
    onView(withId(R.id.add_proof)).perform(click())
    onView(withId(R.id.add_image)).perform(click())
    Espresso.pressBack()
    onView(withId(R.id.add_leader)).perform(click())
    Espresso.pressBack()
    Espresso.pressBack()
    onView(withId(R.id.see_proofs)).perform(click())
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

onView(withText(alert)).check(matches(isDisplayed()))
onView(withText("OK")).perform(click())
onView(withId(R.id.delete_proof)).perform(click())
onView(withText(alert)).check(matches(isDisplayed()))
onView(withText("OK")).perform(click())

private fun addTestRouteLong() {
    val point2 = "Rusinowa Polana"
    val point3 = "Wierch Poroniec"
    val pass2points = "4"
    val time2 = "30"
    val point4 = "Lysa Polana"
    val pass3points = "3"
    val time3 = "20"
    onView(withId(R.id.saveRouteAddProof)).perform(click())
    onView(withId(R.id.add_route_button)).perform(click())
    addTestRouteSection()
    onView(withId(R.id.add_mountain_pass_button)).perform(click())
    onView(withId(R.id.official_button)).perform(click())
    onView(Matchers.allOf(withId(withText(point3)),
    withChild(withText(point2)),
    withChild(withText(pass2points)))).perform(click())
    onView(withId(R.id.time_value)).perform(replaceText(time2))
    onView(withId(R.id.add_route_pass_button)).perform(click())
    onView(withId(R.id.add_mountain_pass_button)).perform(click())
    onView(withId(R.id.official_button)).perform(click())

    onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.scrollToPosition<MountainPassPickAdapter<MountainPassOfficial>.MountainPassItemHolder>(8))
    onView(Matchers.allOf(withId(withText(point3)),
    withChild(withText(point4)),
    withChild(withText(pass3points)))).perform(click())
    onView(withId(R.id.time_value)).perform(replaceText(time3))
    onView(withId(R.id.add_route_pass_button)).perform(click())
    onView(withId(R.id.end_button)).perform(click())
    onView(withId(R.id.choose_date_label)).check(matches(isDisplayed()))
    onView(withId(R.id.save_route_btn)).perform(click())

    @Test
    fun testAddLeaderProof() {
        val leaderId = "1"
        addTestRouteLong()

        onView(withId(R.id.last_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RouteAdapter.RouteItemHolder>(0, click()))
        onView(withId(R.id.edit_proofs)).perform(click())
        onView(withId(R.id.add_proof)).perform(click())
    }
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

onView(withId(R.id.add_leader)).perform(click())
onView(withId(R.id.leader_id)).perform(replaceText(leaderId))
onView(withId(R.id.leader_proof_save)).perform(click())
onView(withId(R.id.see_proofs)).perform(click())
onView(withText("Przodownik:")).check(matches(isDisplayed()))
onView(withId(R.id.close)).perform(click())
onView(withId(R.id.delete_proof)).perform(click())

onView(withId(R.id.proof_list)).perform(RecyclerViewActions.actionOnItemAtPosition<ProofListAdapter.ProofItemHolder>(0, click()))
    onView(withId(R.id.close)).perform(click())
    onView(withText(alert)).check(matches(isDisplayed()))
    onView(withText("OK")).perform(click())
    onView(withId(R.id.see_proofs)).perform(click())
    onView(withText(alert)).check(matches(isDisplayed()))
    onView(withText("OK")).perform(click())}

@Test
fun testAddImageProof() {
    addTestRouteLong()

    onView(withId(R.id.last_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RouteAdapter.RouteItemHolder>(0, click()))
        onView(withId(R.id.edit_proofs)).perform(click())
        onView(withId(R.id.add_proof)).perform(click())
        onView(withId(R.id.add_image)).perform(click())

    onView(withId(R.id.route_section_pick_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RouteSectionsPickAdapter.RouteSectionItemHolder>(0, click()))

    onView(withId(R.id.route_section_pick_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RouteSectionsPickAdapter.RouteSectionItemHolder>(1, click()))

    onView(withId(R.id.pick_picture_button)).check(matches(isDisplayed()))
}

@Test
fun testHasProof() {
    val point1 = "Parking TPN"
    val point2 = "Rusinowa Polana"
    val point3 = "Wierch Poroniec"
    val pass2points = "4"
    val time2 = "30"
    val leaderId = "1"
    onView(withId(R.id.saveRouteAddProof)).perform(click())
    onView(withId(R.id.add_route_button)).perform(click())
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

        addTestRouteSection()
        onView(withId(R.id.add_mountain_pass_button)).perform(click())
        onView(withId(R.id.official_button)).perform(click())
        onView(Matchers.allOf(withChild(withText(point2)),
        withChild(withText(point3)),
        withChild(withText(pass2points)))).perform(click())
            onView(withId(R.id.time_value)).perform(replaceText(time2))
            onView(withId(R.id.add_route_pass_button)).perform(click())
            onView(Matchers.allOf(withChild(withText(point2)),
            withChild(withText(point3)),
            withChild(withText("NIE")))).check(matches(isDisplayed()))
                onView(Matchers.allOf(withChild(withText(point2)),
                withChild(withText(point1)),
                withChild(withText("NIE")))).check(matches(isDisplayed()))
                    onView(withId(R.id.add_proof_button)).perform(click())
                    onView(withId(R.id.add_leader)).perform(click())
                    onView(withId(R.id.leader_id)).perform(replaceText(leaderId))
                    onView(withId(R.id.leader_proof_save)).perform(click())
                    onView(Matchers.allOf(withChild(withText(point2)),
                    withChild(withText(point3)),
                    withChild(withText("TAK")))).check(matches(isDisplayed()))
                        onView(Matchers.allOf(withChild(withText(point2)),
                        withChild(withText(point1)),
                        withChild(withText("TAK")))).check(matches(isDisplayed())))
    }
}

```

Test Results		1 m 19 s	5/5
✓	AddProofFunctionalityTest	1 m 19 s	5/5
✓	testShowLayoutEditRoute	32 s	✓
✓	testShowLayoutNewRoute	8 s	✓
✓	testAddImageProof	13 s	✓
✓	testHasProof	9 s	✓
✓	testAddLeaderProof	15 s	✓

10.4. Potwierdzenie trasy

```

class ConfirmRouteFunctionalityTest {

    @Rule
    @JvmField
    val mActivityRule = ActivityTestRule(MainActivity::class.java, false,
    true)

    @Before
    fun createRouteWithLeaderProof() {
        onView(withId(R.id.saveRouteAddProof)).perform(click())
        onView(withId(R.id.add_route_button)).perform(click())
        onView(withId(R.id.add_mountain_pass_button)).perform(click())
    }
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

onView(withId(R.id.official_button)).perform(click())

onView(withId(R.id.mountain_passes_list_pick)).perform(RecyclerViewActions.actionOnItemAtPosition<MountainPassAdapter.MountainPassOfficialItemHolder>(0, click()))
    onView(withId(R.id.time_value)).perform(replaceText("55"))
    onView(withId(R.id.add_route_pass_button)).perform(click())
    onView(withId(R.id.add_proof_button)).perform(click())
    onView(withId(R.id.add_leader)).perform(click())
    onView(withId(R.id.leader_id)).perform(replaceText("1"))
    onView(withId(R.id.leader_proof_save)).perform(click())
    onView(withId(R.id.end_button)).perform(click())
    onView(withId(R.id.save_route_btn)).perform(click())

onView(withId(R.id.last_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<MountainPassAdapter.MountainPassOfficialItemHolder>(0, click()))
    onView(withId(R.id.send_to_leader)).perform(click())
    onView(withText("OK")).perform(click())
    Espresso.pressBack()
    Espresso.pressBack() }

@Test
fun testShowLayout() {
    val routesToConfirm = "Trasy do potwierdzenia:"
    val confirmRoute = "Potwierdź trasę"
    val routeSections = "Odcinki trasy"
    val proof = "Dowód:"
    val mountainPasses = "Odcinki:"
    val alert = "Uwaga"
    val cancel = "Cofnij"
    onView(withId(R.id.confirmRoute)).check(matches(isDisplayed()))
    onView(withId(R.id.confirmRoute)).perform(click())
    onView(withText(routesToConfirm)).check(matches(isDisplayed()))

    onView(withId(R.id.confirm_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RoutesToConfirmAdapter.RouteItemHolder>(0, click()))
        onView(withText(confirmRoute)).check(matches(isDisplayed()))

    onView(withId(R.id.show_route_sections)).check(matches(isDisplayed()))
        onView(withId(R.id.show_on_map)).check(matches(isDisplayed()))
        onView(withId(R.id.show_route_proofs)).check(matches(isDisplayed()))
        onView(withId(R.id.reject_route)).check(matches(isDisplayed()))
        onView(withId(R.id.confirm_route)).check(matches(isDisplayed()))
        onView(withId(R.id.show_route_sections)).perform(click())
        onView(withText(routeSections)).check(matches(isDisplayed()))

    onView(withId(R.id.close_route_sections)).check(matches(isDisplayed()))
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

onView(withId(R.id.route_sections_small_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RouteSectionSmallAdapter.RouteSectionSmallItemHolder>(0, click()))
    onView(withText(proof)).check(matches(isDisplayed()))

onView(withId(R.id.close_section_details)).check(matches(isDisplayed()))
    onView(withId(R.id.close_section_details)).perform(click())
    onView(withText(routeSections)).check(matches(isDisplayed()))
    onView(withId(R.id.close_route_sections)).perform(click())
    onView(withText(confirmRoute)).check(matches(isDisplayed()))
    onView(withId(R.id.show_on_map)).perform(click())
    onView(withId(R.id.map)).check(matches(isDisplayed()))
    Espresso.pressBack()
    onView(withText(confirmRoute)).check(matches(isDisplayed()))
    onView(withId(R.id.show_route_proofs)).perform(click())
    onView(withText(mountainPasses)).check(matches(isDisplayed()))
    onView(withText(proof)).check(matches(isDisplayed()))
    onView(withId(R.id.close)).check(matches(isDisplayed()))
    onView(withId(R.id.close)).perform(click())
    onView(withId(R.id.reject_route)).perform(click())
    onView(withText(alert)).check(matches(isDisplayed()))
    onView(withText(cancel)).perform(click())
    onView(withId(R.id.confirm_route)).perform(click())
    onView(withText(alert)).check(matches(isDisplayed()))
    onView(withText(cancel)).perform(click())}

@Test
fun testConfirmRoute() {
    val alert = "Uwaga"
    val ok = "OK"
    val routesToConfirm = "Trasy do potwierdzenia:"
    onView(withId(R.id.confirmRoute)).perform(click())

    onView(withId(R.id.confirm_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RoutesToConfirmAdapter.RouteItemHolder>(0, click()))
        onView(withId(R.id.confirm_route)).perform(click())
        onView(withText(alert)).check(matches(isDisplayed()))
        onView(withText(ok)).perform(click())
        onView(withText(routesToConfirm)).check(matches(isDisplayed()))
        Espresso.pressBack()
        onView(withId(R.id.saveRouteAddProof)).perform(click())

    onView(withId(R.id.last_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RouteAdapter.RouteItemHolder>(0, click()))
        onView(withId(R.id.edit_proofs)).check(matches(isNotEnabled()))
        onView(withId(R.id.edit_route)).check(matches(isNotEnabled()))
        onView(withId(R.id.delete_route)).check(matches(isNotEnabled()))
}

```

E-książeczka GOT PTTK	
Projekt	Data: 18.01.2021 r.

```

onView(withId(R.id.send_to_leader)).check(matches(isNotEnabled()))}

@Test
fun testRejectRoute() {
    val alert = "Uwaga"
    val ok = "OK"
    val routesToConfirm = "Trasy do potwierdzenia:"
    onView(withId(R.id.confirmRoute)).perform(click())

    onView(withId(R.id.confirm_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RoutesToConfirmAdapter.RouteItemHolder>(0, click()))
        onView(withId(R.id.reject_route)).perform(click())
        onView(withText(alert)).check(matches(isDisplayed()))
        onView(withText(ok)).perform(click())
        onView(withText(routesToConfirm)).check(matches(isDisplayed()))
    Espresso.pressBack()
    onView(withId(R.id.saveRouteAddProof)).perform(click())

    onView(withId(R.id.last_routes_list)).perform(RecyclerViewActions.actionOnItemAtPosition<RouteAdapter.RouteItemHolder>(0, click()))
        onView(withId(R.id.edit_proofs)).check(matches(isNotEnabled()))
        onView(withId(R.id.edit_route)).check(matches(isNotEnabled()))
        onView(withId(R.id.delete_route)).check(matches(isNotEnabled()))
        onView(withId(R.id.send_to_leader)).check(matches(isNotEnabled()))
    }
}

```

Tests	Duration	po_app
Test Results	1 m 9 s	3/3
ConfirmRouteFunctionalityTest	1 m 9 s	3/3
testConfirmRoute	23 s	✓
testShowLayout	28 s	✓
testRejectRoute	17 s	✓