

1. Spis treści

1. Spis treści	2
2. Opis wymagań	2
3. Diagram przypadków użycia	4
4. Diagram klas analityczny	5
5. Scenariusz	6
6. Diagram aktywności	6
7. Diagram stanów	7
8. Skutki analizy dynamicznej	8
9. Decyzje projektowe	9
10. Diagram klas projektowy	12
11. Koncept GUI	13
12. Wykorzystane technologie	19

2. Opis wymagań

Organizacja wyjazdów grupowych to złożony proces wymagający sprawnej koordynacji wielu aspektów, takich jak komunikacja z uczestnikami wyjazdu, zarządzanie wyjazdem oraz zarządzanie danymi uczestników. Dla agencji podróży kluczowe jest posiadanie narzędzia, które uprości te zadania, zapewniając szybka interakcja z danymi oraz wygodę zarówno dla pracowników, jak i uczestników wyjazdu.

1. Każdy użytkownik posiada swój profil zawierający dane osobowe (imię, nazwisko, numer telefonu, adres e-mail) oraz opcjonalne dane identyfikacyjne. Użytkownik może edytować swoje dane osobowe. Użytkownik komunikuje z systemem w sposób że jest mu widoczna lista wyjazdów organizowanych przez firmy zajmujące się takimi wyjazdami. Użytkownik może zapisać się na taki wyjazd.
2. Dane użytkowników są udostępnione organizatorowi oraz kierownikowi grupy, są oni pracownikami firm organizujących wyjazdy. W systemie są przechowywane dane kontaktowe wszystkich pracowników w tym (Imię, Nazwisko, Numer telefonu i E Mail). Kierownik jest przypisywany przez organizatora do wyjazdu niezależnie od innych uczestników wyjazdu. W jednym wyjeździe jest tylko jeden kierownik wyjazdu. Organizatorzy mogą dodawać, edytować i usuwać wyjazdy. Użytkownicy zapisują się na wyjazd a później są dodawane przez organizatora. Podczas zapisu użytkownika na wyjazd zapisujemy informację o kolejności zapisu użytkownika oraz datę zapisania się. Jest możliwość zrezygnowania z wyjazdu, organizator może dodawać osób z listy zapisanych osób na wyjazd. Organizator też może

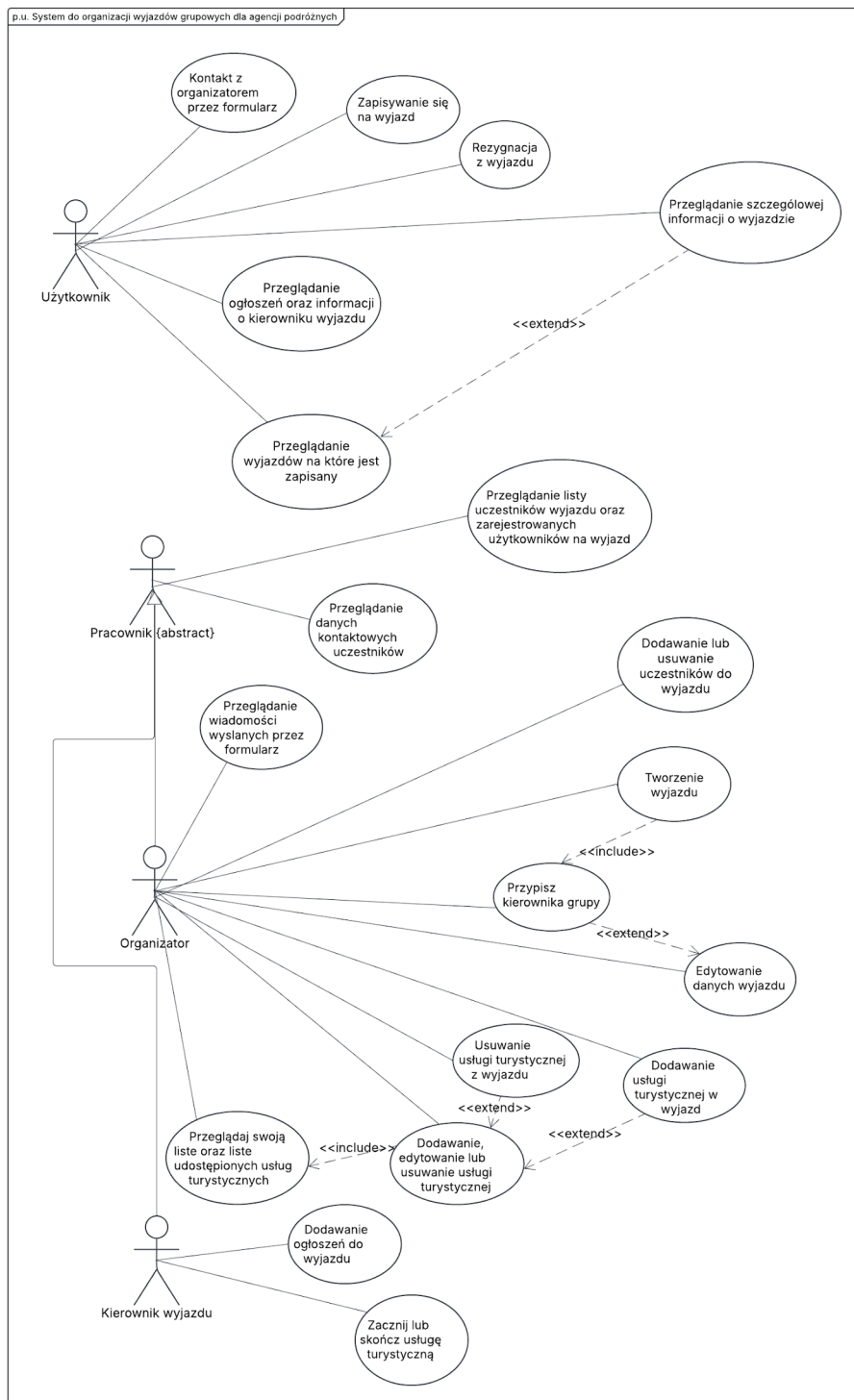
usuwać użytkowników z wyjazdu. Organizator może zarządzać tylko wyjazdami utworzonymi przez niego.

3. Kiedy użytkownik staje uczestnikiem wyjazdu dostaje on możliwość przeglądania ogłoszeń i informacji o kierowniku wyjazdu. Ogłoszenia jest to jedyny sposób komunikacji kierownika z uczestnikami wyjazdu. Zawiera ono datę napisania oraz treść. Użytkownicy są informowane o zmianach w harmonogramie, statusie płatności oraz innych istotnych informacjach. Jest też możliwy kontakt firmą organizującą wyjazdy, jest to możliwe poprzez formularz w systemie. Formularzem są przesyłane takie dane jak wysyłający formularz, odbiorca, data wysłania oraz tekst.
4. Przechowywane także dane firmy są to nazwa oraz email. Pracownicy mogą należeć tylko do jednej firmy. Firma jest organizatorem wyjazdu. Wyjazd zawiera takie informacje jak nazwę, termin naboru uczestników, miejsce docelowe, miejsce wyjazdu, data początku, data końca, opis programu wyjazdu, liczba osób w grupie, cena oraz status wyjazdu {"W trakcie naboru", "Skończył się"}.
5. Podczas tworzenia wyjazdu organizator może również podać informacje o dodatkowych usługach turystycznych. Są to informacje dotyczące transportu oraz miejsc noclegowych, do jednego wyjazdu jest przypisany co najmniej jeden transport i hotel. Każda usługa turystyczna przechowuje nazwę oraz numer telefonu. Dla transportu jest dodatkowo zapisywany jego rodzaj oraz opcjonalnie firma przewoźnika. A dla hotelu adres oraz opcjonalnie strona internetowa i email. W ramach wyjazdu, dla każdego środka transportu przypisanego do wyjazdu, system zapisuje miejsce wyjazdu, datę wyjazdu oraz datę powrotu. W przypadku noclegu, system przechowuje daty zameldowania i wymeldowania uczestników oraz na podstawie tych dat wylicza czas pobytu. Wszystkie usługi turystyczne są zarządzane przez organizatora ale ich stanem w czasie podróży zarządza kierownik grupy.

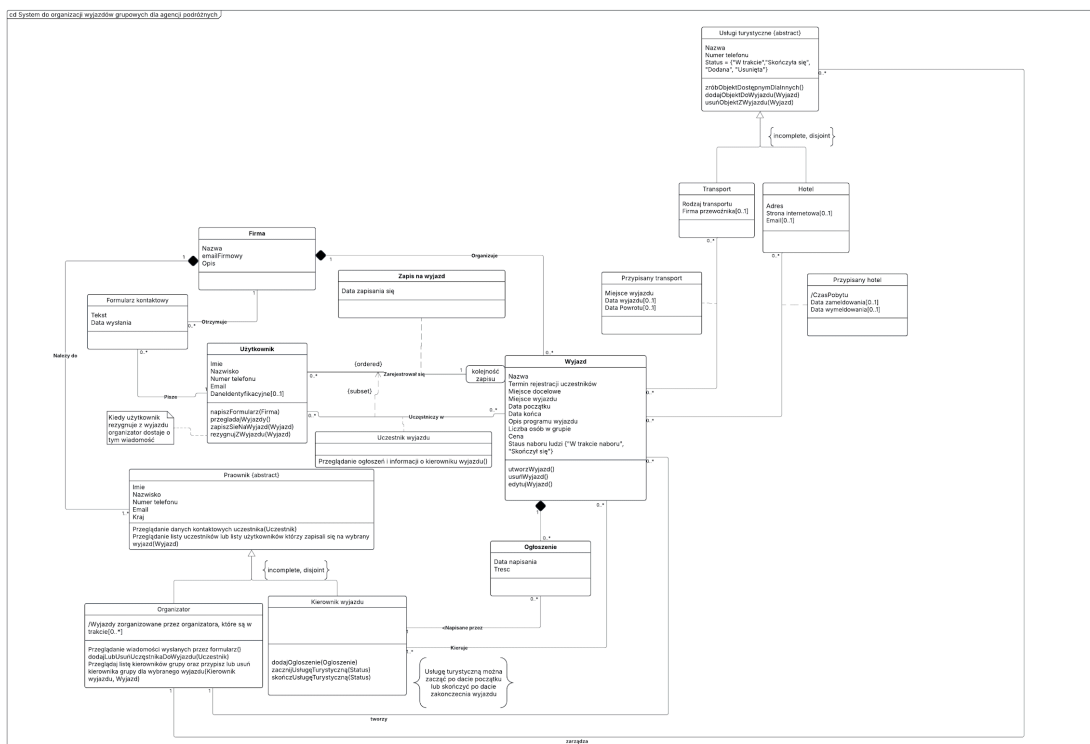
Podział funkcji według ról użytkowników

- **Uczestnik:** Przegląd wyjazdów, zapisanie się/rezygnacja z wyjazdu, przegląd powiadomień, kontakt z organizatorem, przegląd informacji o organizatorze/kierowniku wyjazdu oraz przegląd wyjazdu szczegółowo.
- **Organizator:** Tworzenie, zarządzanie oraz usuwanie wyjazdów, dodawanie uczestników, przypisanie kierownika grupy, zarządzanie usługą turystyczną wyjazdu, komunikacja z uczestnikami i zarządzanie uczestnikami wyjazdu.
- **Kierownik wyjazdu:** Utworzenia ogłoszenia na stronie wyjazdu oraz przegląd informacji o uczestnikach wyjazdu.

3. Diagram przypadków użycia



4. Diagram klas analityczny

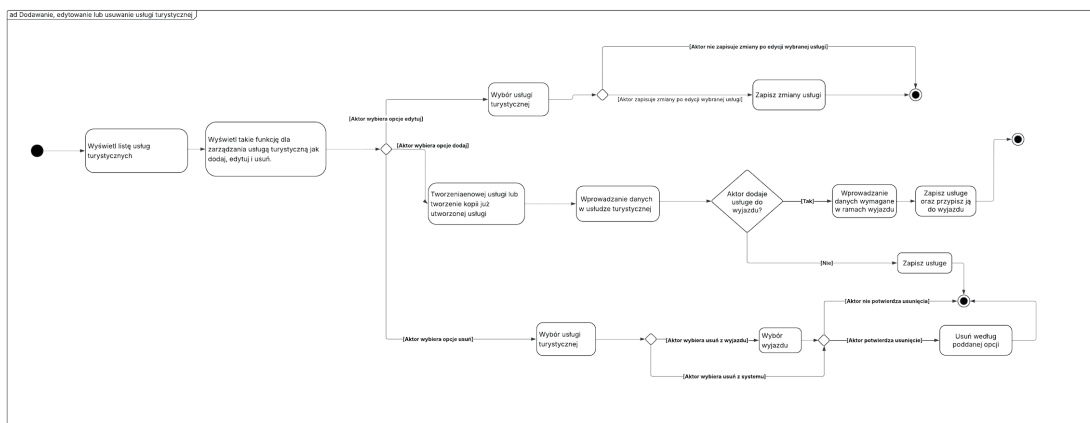


5. Scenariusz

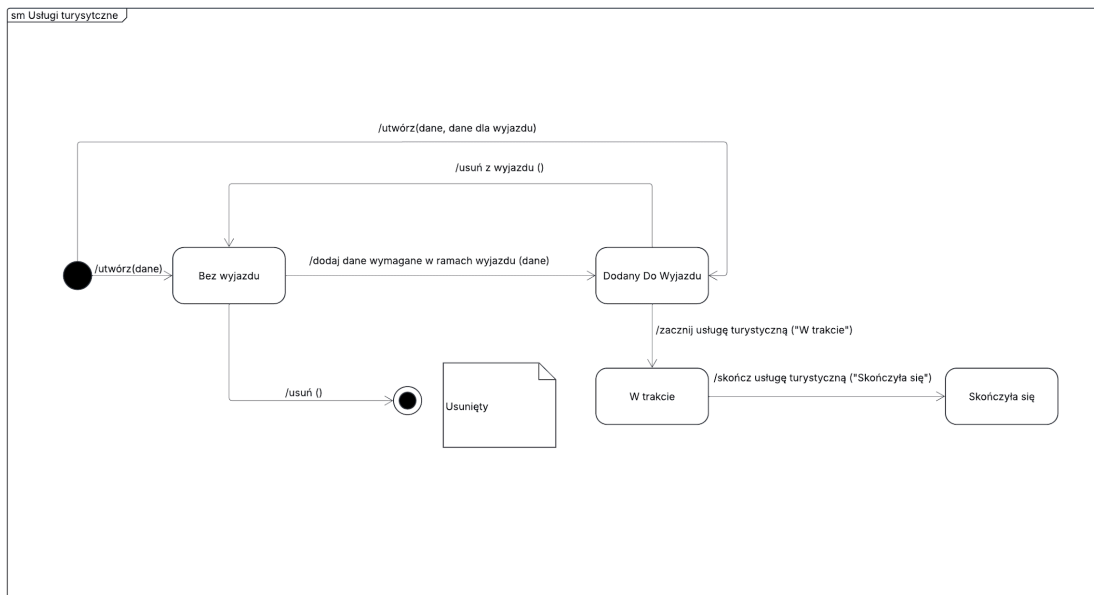
USc Dodawanie, edytowanie lub usuwanie usługi turystycznej

Aktor	Organizator
Warunek początkowy	Ogranizator jest w trakcie przypadku użycia "Przeglądaj swoją listę oraz listę udostępnionych usług turystycznych"
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor uruchamia przypadek użycia. 2. System wyświetla listę wszystkich usług turystycznych dostępnych dla organizatora oraz takie funkcję jak dodaj nową usługę, edytuj, usuń. Aktor wybiera opcję edytuj. 3. System wyświetla wszystkie usługi turystyczne dostępne aktorowi z możliwością ich edycji. Aktor wybiera usługę oraz zapisuje zmiany po edycji usługi. 4. System zapisuje zmiany usługi.
Scenariusz alternatywny	<p>2a Aktor wybiera opcję dodaj nową usługę. System wyświetla funkcje dla utworzenia nowej usługi jako kopii już utworzonej.</p> <p>2aa Aktor wybiera opcję. System prosi o wprowadzenie danych dla usług turystycznych oraz o podanie danych wymaganych dla usługi turystycznej.</p> <p>2aaa Aktor dodaje usługę do wyjazdu i na podstawie typu podaje wszystkie dane. System prosi o podanie danych wymaganych w ramach wyjazdu.</p> <p>2aaaa Aktor podaje dane wymagane w ramach wyjazdu. System zapisuje usługę oraz przypisuje ją do wyjazdu.</p> <p>2aaab Aktor nie podaje danych. System zapisuje usługę oraz kończy swoją pracę.</p> <p>2b Aktor wybiera opcję usuń oraz wybiera usługę turystyczną. System wyświetla opcje usunięcia usługi turystycznej.</p> <p>2ba Aktor wybiera usuń z wyjazdu. System prosi o wybór wyjazdu.</p> <p>2baa Aktor wybiera wyjazd. System prosi o potwierdzenie usunięcia według podanej opcji.</p> <p>2baaa Aktor potwierdza. System usuwa usługę oraz kończy swoją pracę.</p> <p>2baab Aktor nie potwierdza. System kończy swoją pracę.</p> <p>2bb Aktor wybiera usuń z systemu. System prosi o potwierdzenie usunięcia według podanej opcji.</p> <p>2bba Aktor potwierdza. System usuwa usługę oraz kończy swoją pracę.</p> <p>2bbb Aktor nie potwierdza. System kończy swoją pracę.</p> <p>3a Aktor nie zapisuje zmiany po edycji. System kończy swoją pracę.</p>
Warunek końcowy	Brak

6. Diagram aktywności



7. Diagram stanów



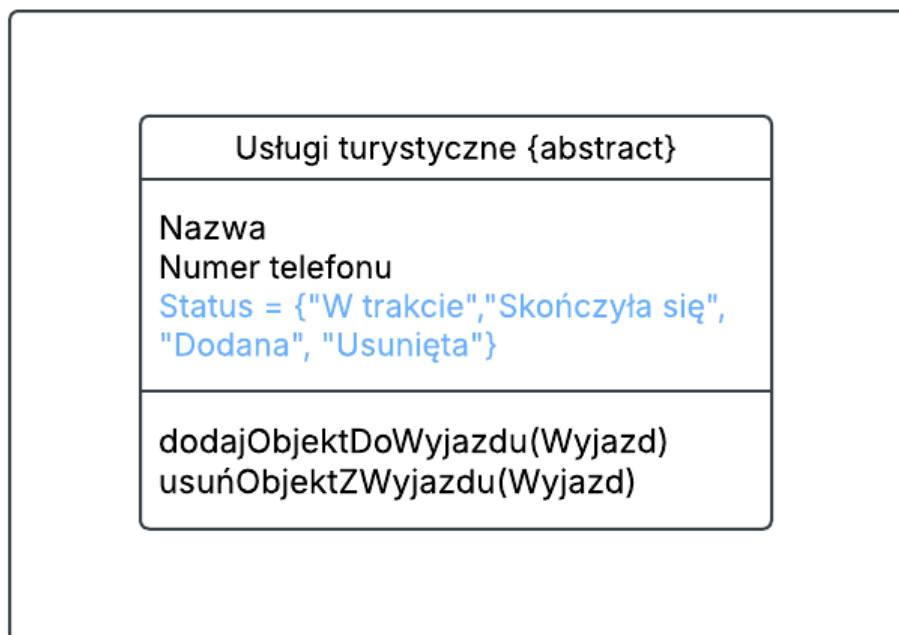
8. Skutki analizy dynamicznej

Podczas analizy dynamicznej była podjęta decyzja że usługi turystyczne powinny przechowywać ich status w systemie jako atrybut w klasie usługi turystyczne.

Jest to istotna rzecz dla usług turystycznych ponieważ są one zarządzane przez dwóch pracowników, organizatora i kierownika grupy.

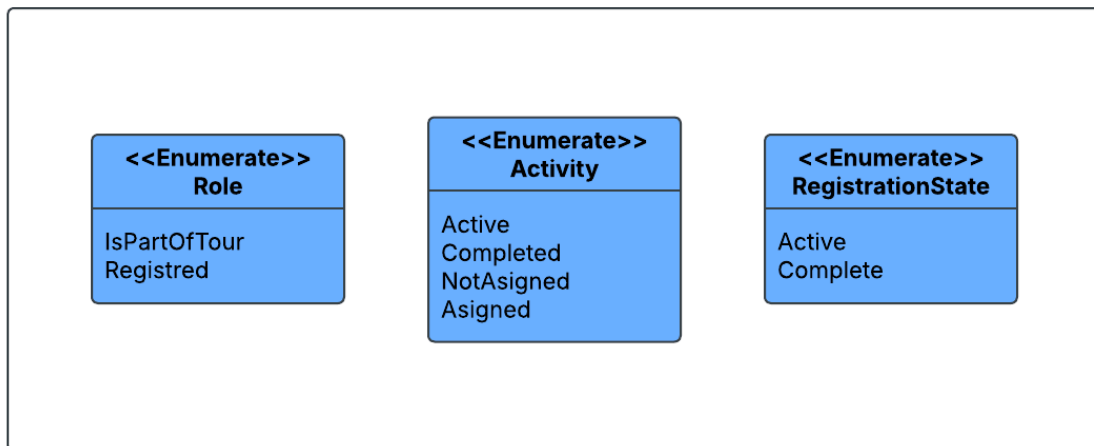
Za pomocą tego atrybutu zmieniamy jego dostępność dla pracownika w zależności od statusu w którym on jest.

Organizator może zarządzać usługą tylko kiedy jest to statusy "Bez Wyjazdu" lub "Dodany Do Wyjazdu" a kierownik grupy "Dodany Do Wyjazdu" oraz "W trakcie".

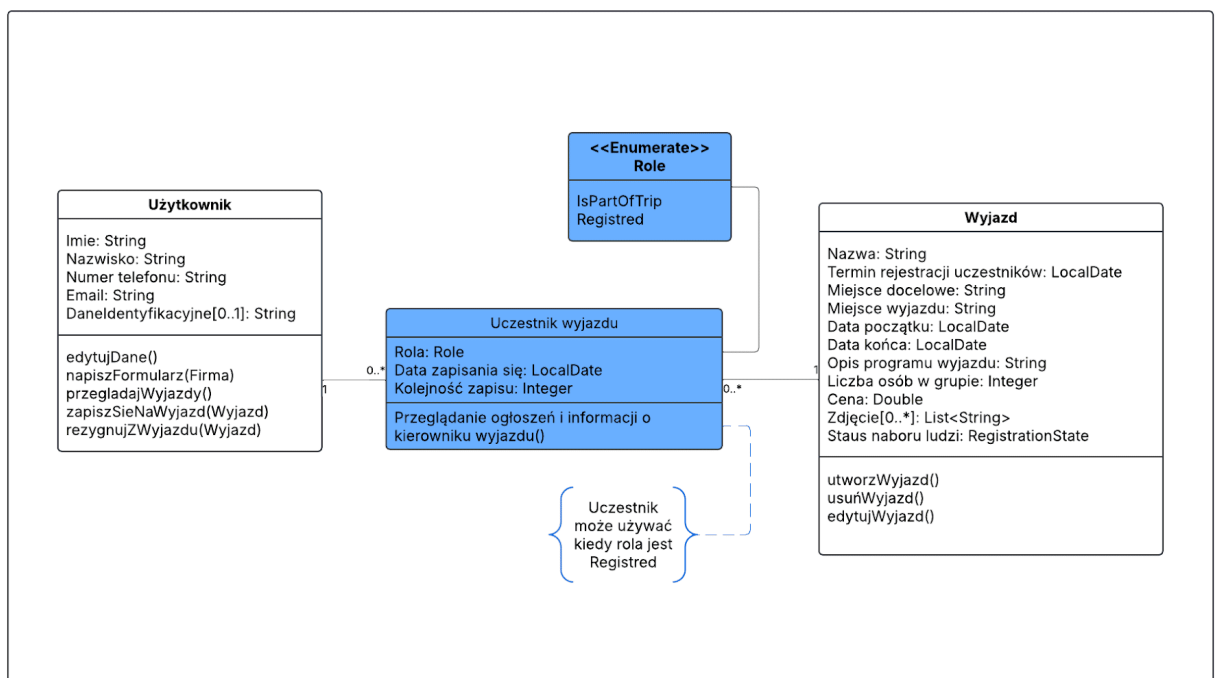
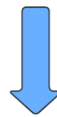
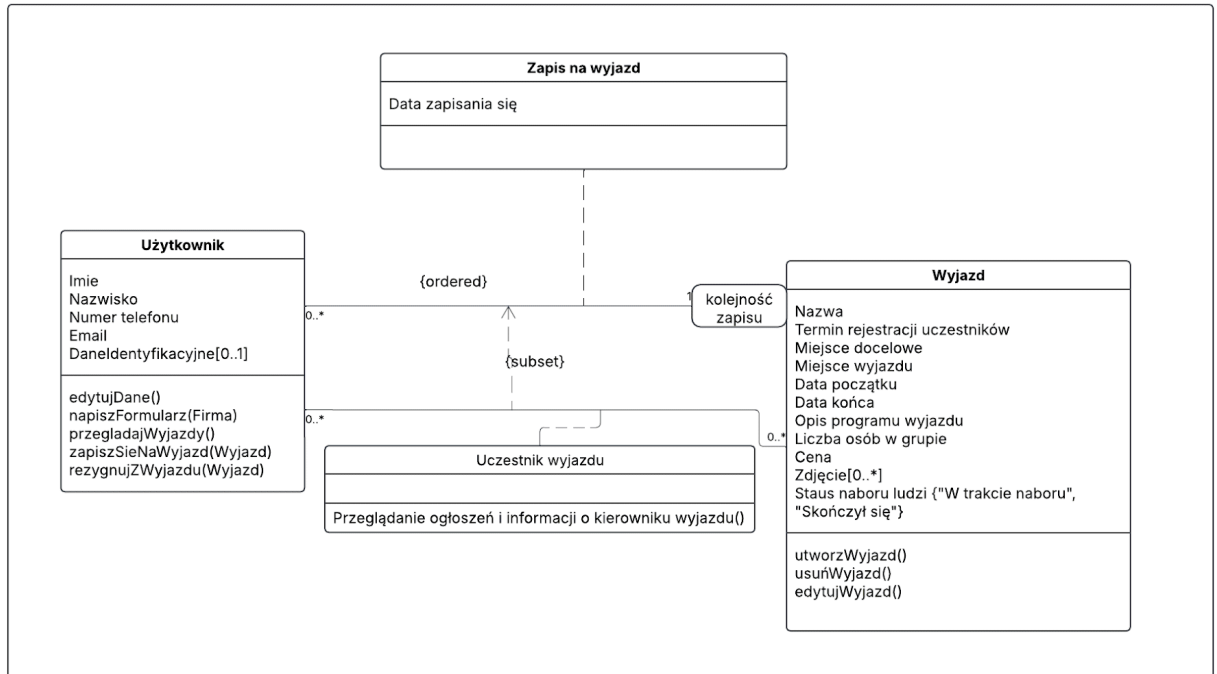


9. Decyzje projektowe

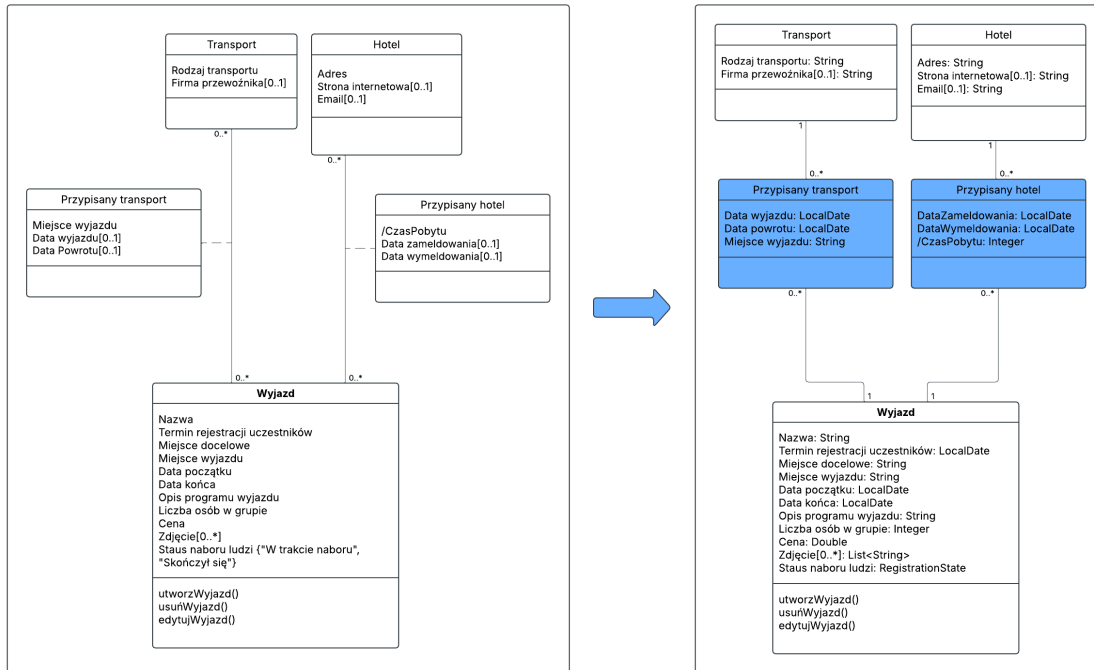
- Kompozycja została zaimplementowana używając usunięcia kaskadowego.
- Trwałość danych jest zachowywana poprzez użycie bazy danych PostgreSQL oraz technologii Spring Data JPA.
- Wszystkie atrybuty, które mają predefiniowane kategorię zostały zaimplementowane jako ENUM:



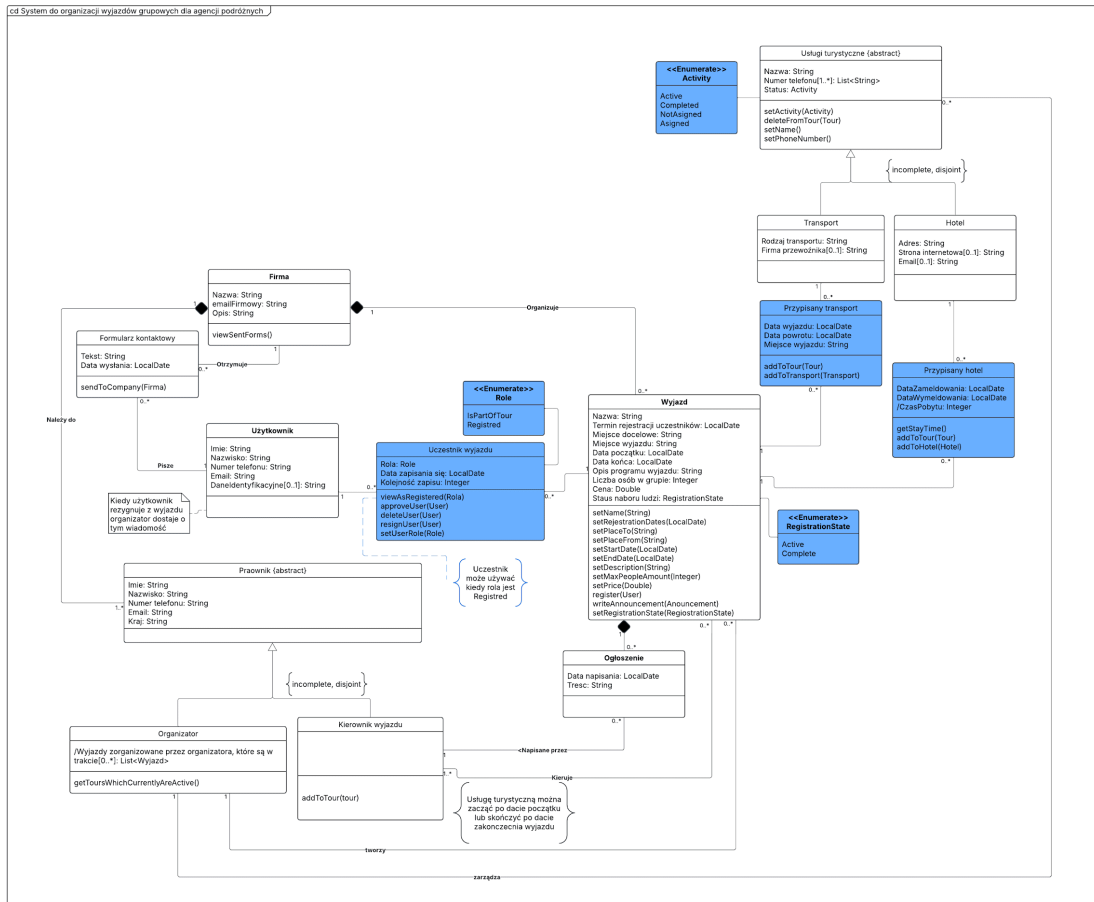
- Podczas implementacji ograniczenia {subset} była podjęta decyzja spłaszczenie asocjacji z atrybutami w jedną klasę.



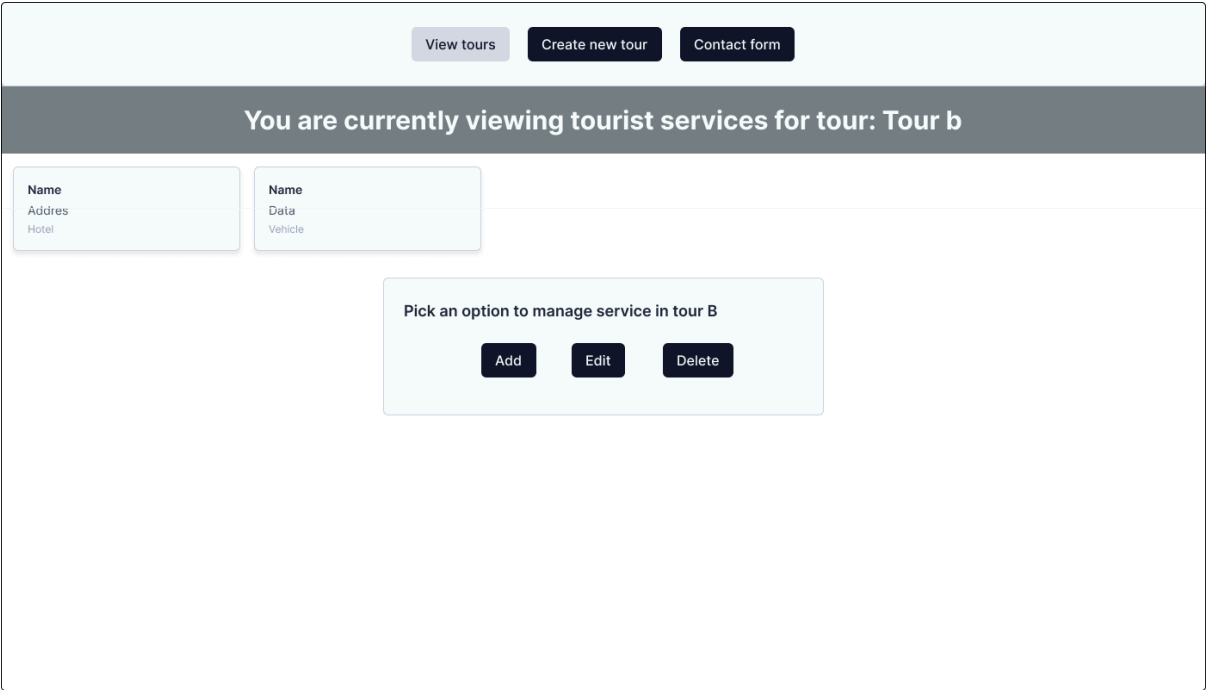
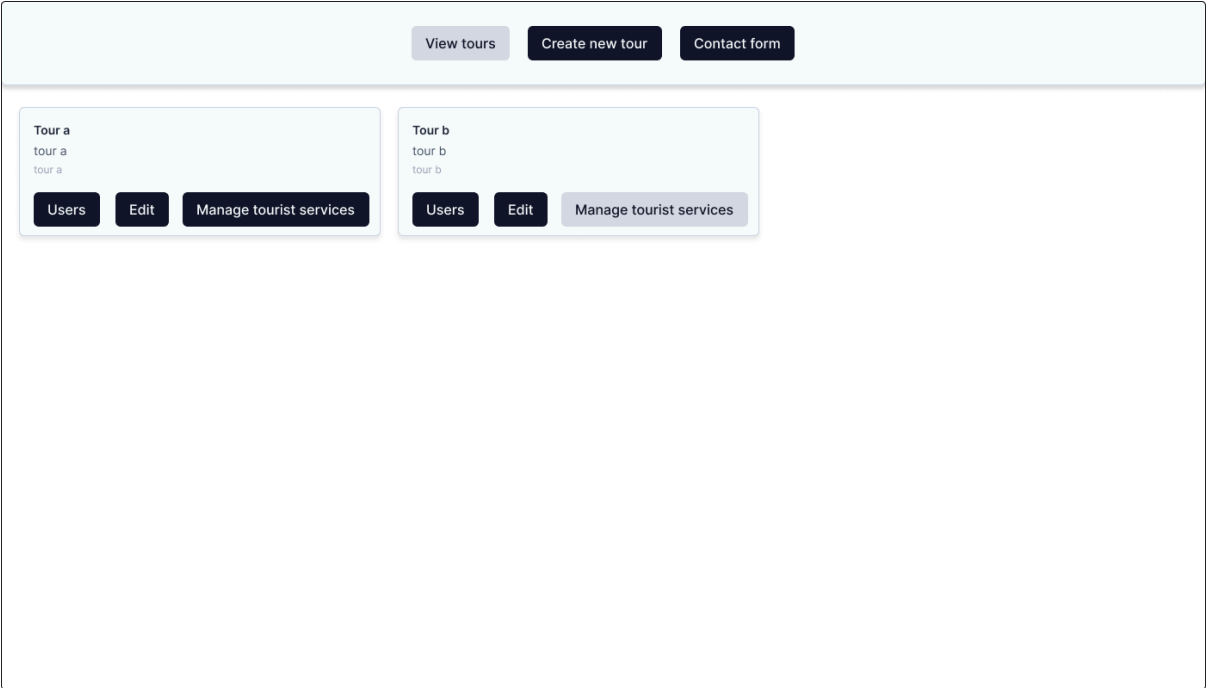
- Wszystkie asocjacje z atrybutem są implementowane jako dodatkowa klasa połączona z innymi za pomocą relacji 1..*:



10. Diagram klas projektowy



11. Konzept GUI



Edytowanie usługi turystycznej:

View tours

Create new tour

Contact form

You are currently viewing tourist services for tour: Tour b

Name

Address

Hotel

Name

Data

Vehicle

View tours

Create new tour

Contact form

You are currently viewing tourist services for tour: Tour b

Name

Address

Hotel

Name

Data

Vehicle

Edit an Name tourist service

Name

Phone Number

Hotel/ Vehicle

Address

Web Page

Email

Cancel

Save changes

Dodawanie usługi turystycznej:

[View tours](#) [Create new tour](#) [Contact form](#)

You are currently viewing tourist services for tour: Tour b

Name
Address
Hotel

Name
Data
Vehicle

+

[View tours](#) [Create new tour](#) [Contact form](#)

You are currently viewing tourist services for tour: Tour b

Name
Address
Hotel

Name
Data
Vehicle

Create new tourist service

Name

Phone Number

Hotel/
Vehicle

Address

Web Page

Email

Cancel

Save changes

View tours

Create new tour

Contact form

You are currently viewing tourist services for tour: Tour b

Name

Address

Hotel

Name

Data

Vehicle

Add service Name

Choose tour

Start date

End date

Cancel

Save changes

Usuwanie usługi turystycznej:

[View tours](#) [Create new tour](#) [Contact form](#)

You are currently viewing tourist services for tour: Tour b

Name
Address
Hotel

Name
Data
Vehicle

[View tours](#) [Create new tour](#) [Contact form](#)

You are currently viewing tourist services for tour: Tour b

Name
Address
Hotel

Name
Data
Vehicle

Name would be deleted from:
[Tour](#) [System](#)

View tours

Create new tour

Contact form

You are currently viewing tourist services for tour: Tour b

Name

Address

Hotel

Name

Data

Vehicle

Choose tour from which you want delete service

Tour

Delete

View tours

Create new tour

Contact form

You are currently viewing tourist services for tour: Tour b

Name

Address

Hotel

Name

Data

Vehicle

Are you sure that you want to delete?

Yes

Cancel

12. Wykorzystane technologie

- **Spring Boot** - został wybrany jako główna technologia systemu. Oferuje on wszystkie potrzebne funkcje dla utworzenia aplikacji webowej oraz jego przynależność do ekosystemu **Spring** pozwala nam łatwą integrację z innymi komponentami **Spring**, tacyśmy jak **Spring Jpa Data**.
 - **Maven** - jako narzędzie dla automatyzacji procesu tworzenia oprogramowania jest używane w celu uproszczenia struktury projektu oraz zmniejszenie liczby kodu wymaganego dla systemu.
 - **Spring Jpa Data** - wybór tej technologii jest spowodowany tym że jest ona ściśle zintegrowany z inną technologią **Hibernate**. **Hibernate** pozwala mapowanie obiektów w kodzie według utworzonego diagramu zapewniając to że nasze dane są przechowywane i zarządzane w ramach zasad UML-a. W połączeniu z tym inne możliwości, które oferuje **Spring Jpa Data**, pozwalają nam utrzymywanie naszego kodu czystym oraz minimalistycznym.
 - **PostgreSQL** - jest to baza danych dla zapewnienia trwałości systemu. Była wybrana z powodu jej łatwej integracji w ekosystem **Spring** oraz ze względu na jej skalowalność oraz elastyczność.
 - **React** - Wybór tej technologii jako interfejsu aplikacji był spowodowany tym że react jest oparty na komponentach. Użycia takich komponentów pozwala wielokrotnie używać napisany kod tym samym polepszając czytelność oraz strukturę kodu.
1. Make all tourist services visible to organiser that he have rights to
 - a. Probably remake so it would be organiser -> tourist services -> trips
 2. Delete all console logs
 3. Clean css
 4. Make all functions from uml and stuff from it