

System obsługi wycieczek turystycznych

inż. Marek Borzyszkowski 184266

inż. Jakub Czermiński 184543

inż. Agata Geisler 184780

inż. Krzysztof Kuchta 184568

Spis treści:

1. Podział na komponenty i wskazanie agregatów
2. Schematy baz danych komponentów i CQRS
3. Kontrakty i Event sourcing
4. Sagi
5. Opis scrappera
6. Diagram klas i metod

1. Podział na komponenty i wskazanie agregatów

Komponenty:

Wycieczka

- komponent odpowiedzialny za generowanie konkretnych wycieczek w zależności od określonych parametrów przy odpowiednim sprawdzeniu ich dostępności;
- umożliwia wyświetlanie zarówno listy wycieczek jak i pojedynczej wygenerowanej a później wybranej wycieczki;
- Kryteria_Użytkownika: { Ile osób, Kraj, Skąd, Terminy wycieczki Od/Do } - kryteria wskazane przez użytkownika według których generowana będzie lista wycieczek do wyświetlenia na stronie;
- Kryteria_Wygenerowane: { Id_Hotelu, Ilość osób, Kraj, Termin wycieczki Od/Do } - kryteria wskazanej przez użytkownika wycieczki, wybranej z wcześniej wygenerowanej listy wycieczek, według których generowana jest wycieczka do wyświetlenia na stronie.

Hotel

- komponent przechowujący informacje o hotelach dostępnych w konkretnych lokalizacjach;
- Kryteria_Hoteli: { Kraj, Terminy Od/Do, Ilość Osób};
- Kryteria_Hotelu: { Ile osób, Terminy Od/Do, Hotel_Id};

Transport

- komponent przechowujący informacje o transporcie dostępnym w ramach dojazdu i/lub powrotu z konkretnej lokalizacji;

- Kryteria_Transportów: { Skąd, Ile osób, Terminy Od/Do, Kraj };
- Kryteria_Transportu: { Terminy Od/Do, Ile osób, Kraj }.

Oferta

- komponent będący konkretną konfiguracją wycieczki w zależności od określonych parametrów w momencie rezerwacji przez użytkownika.
- Propozycja Kuby: Komponent zarządzający rezerwacjami wycieczek. Odpowiada za komunikację z komponentami odpowiedzialnymi za rezerwację konkretnych wartości, pilnowanie czasu na rezerwację, a także zwracanie do użytkownika informacji o statusie rezerwacji.

Płatność

- komponent odpowiedzialny za umożliwienie opłacenia oferty, a tym samym przekazywanie jej informacji odnośnie powodzenia dokonania transakcji.

Logowanie

- komponent umożliwiający potencjalnym użytkownikom logowanie do systemu, dzięki czemu możliwa będzie weryfikacja oraz odpowiednie rezerwowanie utworzonej w zależności od kryteriów oferty wycieczki.

Brama

- komponent odpowiedzialny za przekazywanie informacji zarówno od jak i do użytkownika w sposób synchroniczny;
- przekazuje dalej zapytania i komendy do usług, a następnie otrzymuje informacje zwrotne na odpowiednie endpointy.

Agregatory:

paczki danych o hotelach o transporcie, co się dzieje z rezerwacją (?)

2. Schematy baz danych komponentów i CQRS

Hotel:

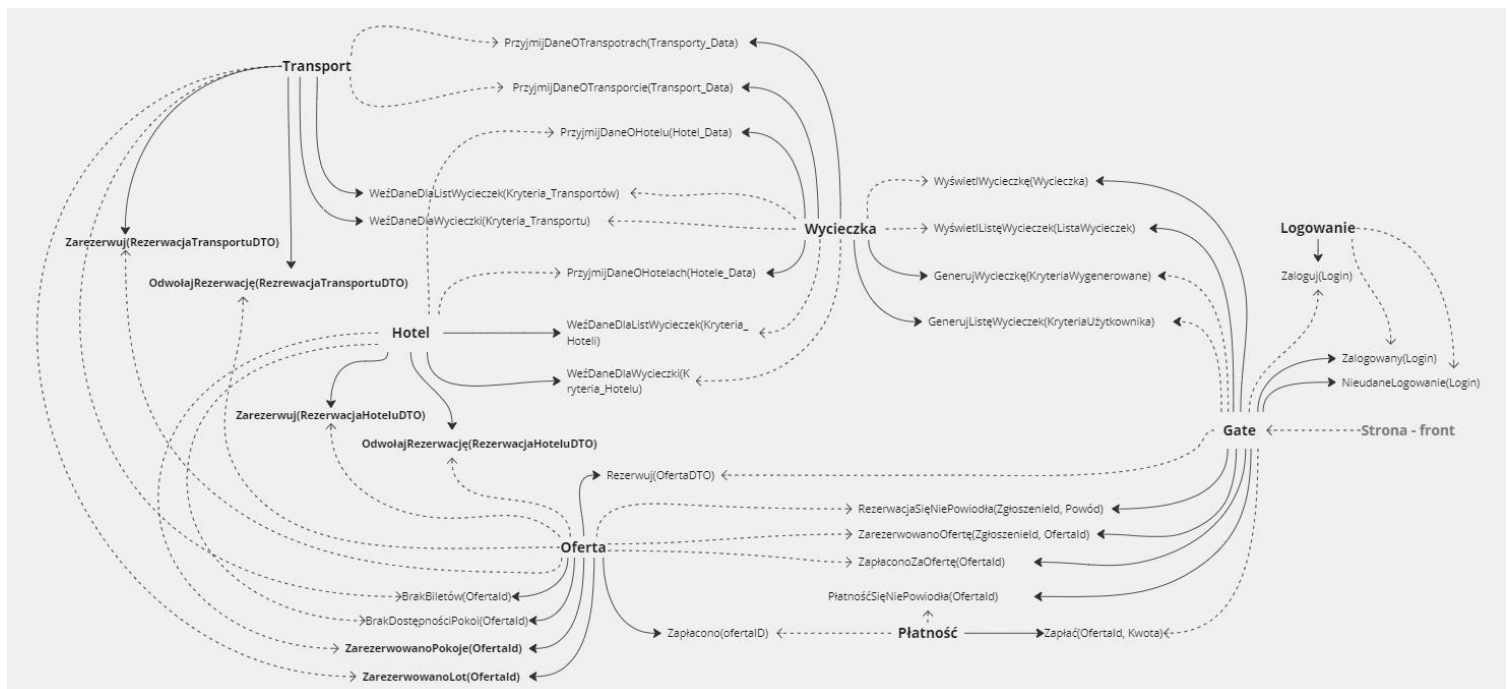
- Hotel:
 - Id
 - Name
 - Country
 - City
 - Discount (wartość 0-1)
- Diet:
 - Id
 - Name

- Hotel_Diet:
 - HotelId
 - DietId
- RoomType:
 - Id
 - Name
- Hotel_RoomType:
 - Id
 - HotelId
 - RoomTypeId
 - NumberOfRooms
 - PriceForTheNight
- Reservation:
 - Id
 - Hotel_RoomTypeId
 - From
 - To
 - NumberOfRooms
- User:
 - User
 - Login
- Transport:
 - Connection
 - Id
 - DeparturePlace
 - ArrivalPlace
 - TravellInstance
 - Id
 - ConnectionId
 - DepartureDate
 - NumberOfSeats
 - Tictet
 - Id
 - TravellInstanceId
 - SeatsTaken
- Offer:
 - Offer
 - Id
 - DepartureTicketId
 - ArrivalTicketId
 - DateFrom

- DateTo
- HotelId
- NumberOfAdults
- NumberOfNewborns
- NumberOfChildren
- NumberOfTeenagers
- UserLogin
- OfferStatus

3. Kontrakty i Event sourcing

Kontrakty:



Wycieczka:

- *PrzyjmijDaneOHotelach(HoteleDTO)*
 - komunikacja Wycieczka -> Hotel;
 - endpoint wycieczki na który komponent Hotel wystawia dane dotycząca hoteli.
- *PrzyjmijDaneOHotelu(HotelDTO)*
 - komunikacja Wycieczka -> Hotel;
 - endpoint wycieczki na który komponent Hotel wystawia dane dotycząca pojedynczego hotelu.
- *PrzyjmijDaneOTranspotrach(TransportyDTO)*
 - komunikacja Wycieczka -> Transport;
 - endpoint wycieczki na który komponent Transport wystawia dane dotyczące transportów.
- *PrzyjmijDaneOTransporcie(TransportDTO):*
 - komunikacja Wycieczka -> Transport;

- endpoint wycieczki na który komponent Transport wystawia dane dotyczące pojedynczego transport.
- *GenerujListęWycieczek(Kryteria_Użytkownika)*
 - komunikacja Wycieczka -> Agregator;
 - endpoint wycieczki na który komponent Agregator wystawia dane dotyczące kryteriów zapewnionych przez użytkownika.
- *GenerujWycieczkę(Kryteria_Wygenerowane)*
 - komunikacja Wycieczka -> Agregator;
 - endpoint wycieczki na który komponent Agregator wystawia dane dotyczące kryteriów wskazanej przez użytkownika wycieczki.

Hotel:

- *WeźDaneDlaListWycieczek(Kryteria_Hoteli)*
 - komunikacja Hotel -> Wycieczka;
 - endpoint hotelu na który komponent Wycieczka wystawia kryteria dotyczące hoteli dla pożądanej listy wycieczek.
- *WeźDaneDlaWycieczki(Kryteria_Hotelu)*
 - komunikacja Hotel -> Wycieczka;
 - endpoint hotelu na który komponent Wycieczka wystawia kryteria dotyczące hoteli dla pożądanej wycieczki.
- *OdwołajRezerwację(RezerwacjaTransportuDTO)*
 - komunikacja Hotel -> Oferta;
 - endpoint hotelu na który komponent Oferta wystawia dane dotyczące odwołania rezerwacji zasobów hoteli.
- *Zarezerwuj(RezerwacjaTransportuDTO)*
 - komunikacja Hotel -> Oferta;
 - endpoint hotelu na który komponent Oferta wystawia dane dotyczące rezerwowanego zasoby hoteli.

Transport:

- *WeźDaneDlaListWycieczek(Kryteria_Transportów)*
 - komunikacja Transport-> Wycieczka;
 - endpoint transportu na który komponent Wycieczka wystawia kryteria dotyczące transportu dla pożądanej listy wycieczek.
- *WeźDaneDlaWycieczki(Kryteria_Transportu)*
 - komunikacja Transport-> Wycieczka;
 - endpoint transportu na który komponent Wycieczka wystawia kryteria dotyczące transportu dla pożądanej wycieczki.
- *OdwołajRezerwację(RezerwacjaHoteluDTO)*
 - komunikacja Transport-> Oferta;
 - endpoint transportu na który komponent Oferta wystawia dane dotyczące odwołania rezerwacji zasobów transportu.
- *Zarezerwuj(RezerwacjaHoteluDTO)*
 - komunikacja Transport-> Oferta;

- endpoint transportu na który komponent Oferta wystawia dane dotyczące rezerwowanego zasoby transportu.

Oferta:

- *Rezerwuj(OfertaId)*
 - komunikacja Oferta -> Brama;
 - endpoint oferty na który komponent Brama wystawia Id oferty, która powinna zostać zarezerwowana na potrzeby płatności.
- *Zapłacono(OfertaId)*
 - komunikacja Oferta -> Płatność;
 - endpoint oferty na którym komponent Płatność wystawia id oferty, która została opłacona przez użytkownika.
- *ZarezerwowanoLot(OfertaId)*
 - komunikacja Oferta -> Transport;
 - endpoint oferty na którym Transport wystawia id oferty dla której lot został już zarezerwowany.
- *BrakBiletów(OfertaId)*
 - komunikacja Oferta -> Transport;
 - endpoint oferty na którym Transport wystawia id oferty jeśli w momencie rezerwacji transportu wystąpiła sytuacja braku biletu.
- *ZarezerwowanoPokoje(OfertaId)*
 - komunikacja Oferta -> Hotel;
 - endpoint oferty na którym Hotel wystawia id oferty dla której pokoje zostały już zarezerwowany.
- *BrakDostępnościPokoji(OfertaId)*
 - komunikacja Oferta -> Hotel;
 - endpoint oferty na którym Hotel wystawia id oferty jeśli w momencie rezerwacji hotelu wystąpiła sytuacja braku dostępności pokoju.

Płatność:

- *Zapłać(Oferta id, Kwota)*
 - komunikacja Płatność -> Brama;
 - endpoint płatności na którym Brama wystawia id oferty oraz kwotę dla oferty którą użytkownik wybrał do opłacenia.

Logowanie:

- *Zaloguj(Login)*
 - komunikacja Logowanie -> Brama
 - endpoint logowania na który Brama wystawia login użytkownika by potwierdzić jego tożsamość.

Brama:

- WyświetlWycieczkę(WycieczkaDTO)
 - komunikacja Brama -> Wycieczka;
 - endpoint bramy na który Wycieczka wystawia dane dotyczące wycieczki, która ma zostać wyświetlona.
- WyświetlListęWycieczek(WycieczkiDTO)
 - komunikacja Brama -> Wycieczka;
 - endpoint bramy na który Wycieczka wystawia dane dotyczące listy wycieczek, która ma zostać wyświetlona.
- RezerwacjaSięNiePowiodła(ZgłoszenieRezerwacjiDTO)
 - komunikacja Brama -> Oferta;
 - endpoint bramy na który Oferta wystawia informację odnośnie rezerwacji jakiego zgłoszenie się nie powiodła.
- ZarezerwowanoOfertę(ZgłoszenieRezerwacjiDTO, OfertaId)
 - komunikacja Brama -> Oferta;
 - endpoint bramy na który Oferta wystawia informację odnośnie rezerwacji i id oferty której rezerwacja się powiodła.
- ZapłaconoZaOfertę(OfertaId)
 - komunikacja Brama -> Oferta;
 - endpoint bramy na który Oferta wystawia informację odnośnie id oferty której zapłacenie się powiodło.
- PłatnośćSięNiePowiodła(OfertaId)
 - komunikacja Brama -> Płatność;
 - endpoint bramy na który Płatność wystawia id oferty, dla którego procedura płatności nie powiodła się.
- Zalogowany(Login)
 - komunikacja Brama -> Logowanie;
 - endpoint bramy na który Logowanie wystawia login użytkownika, dla którego procedura logowania powiodła się.
- NieudaneLogowanie(Login)
 - komunikacja Brama -> Logowanie;
 - endpoint bramy na który Logowanie wystawia login użytkownika, dla którego procedura logowania nie powiodła się.

DTO

WycieczkaDTO: {

Nazwa hotelu

Kraj, miasto

Lista pokoi

Wybrany Lot od/do

Przedział dat

PokojeDTO

Rodzaje posiłków

Możliwe Lotniska od/do - cena per lotnisko dla pojedynczej osoby

};

```
WycieczkiDTO: WycieczkaDTO[];
```

```
HotelDTO: {  
    Nazwa Hotelu,  
    Kraj, Miasto,  
    Przedział Dat,  
    Rodzaje posiłków,  
    PokojeDTO
```

```
};
```

```
HoteleDTO: HotelDTO[];
```

```
RezerwacjaHoteluDTO: {  
    HotelId  
    DataOd  
    DataDo  
    PokojeDTO  
    OfertaId
```

```
}
```

```
PokojDTO: {  
    IlośćPokoi  
    TypPokoju
```

```
}
```

```
PokojeDTO: PokojDTO[]
```

```
TransportDTO: {  
    Wybrany Lot Od/Do,  
    Przedział dat,  
    Możliwe Lotniska od/do
```

```
};
```

```
TransportyDTO: TransportDTO[];
```

```
RezerwacjaTransportuDTO: {  
    PodróżId  
    Liczba  
    OfertaId
```

```
}
```

```
OfertaDTO: {  
    Nazwa Hotelu  
    Kraj, Miasto  
    Wybrany Lot Od/Do  
    Przedział Dat  
    Wybrane Pokoje  
    Wybrany rodzaj posiłków  
    Lotnisko  
    Ilość osób
```

```
}
```

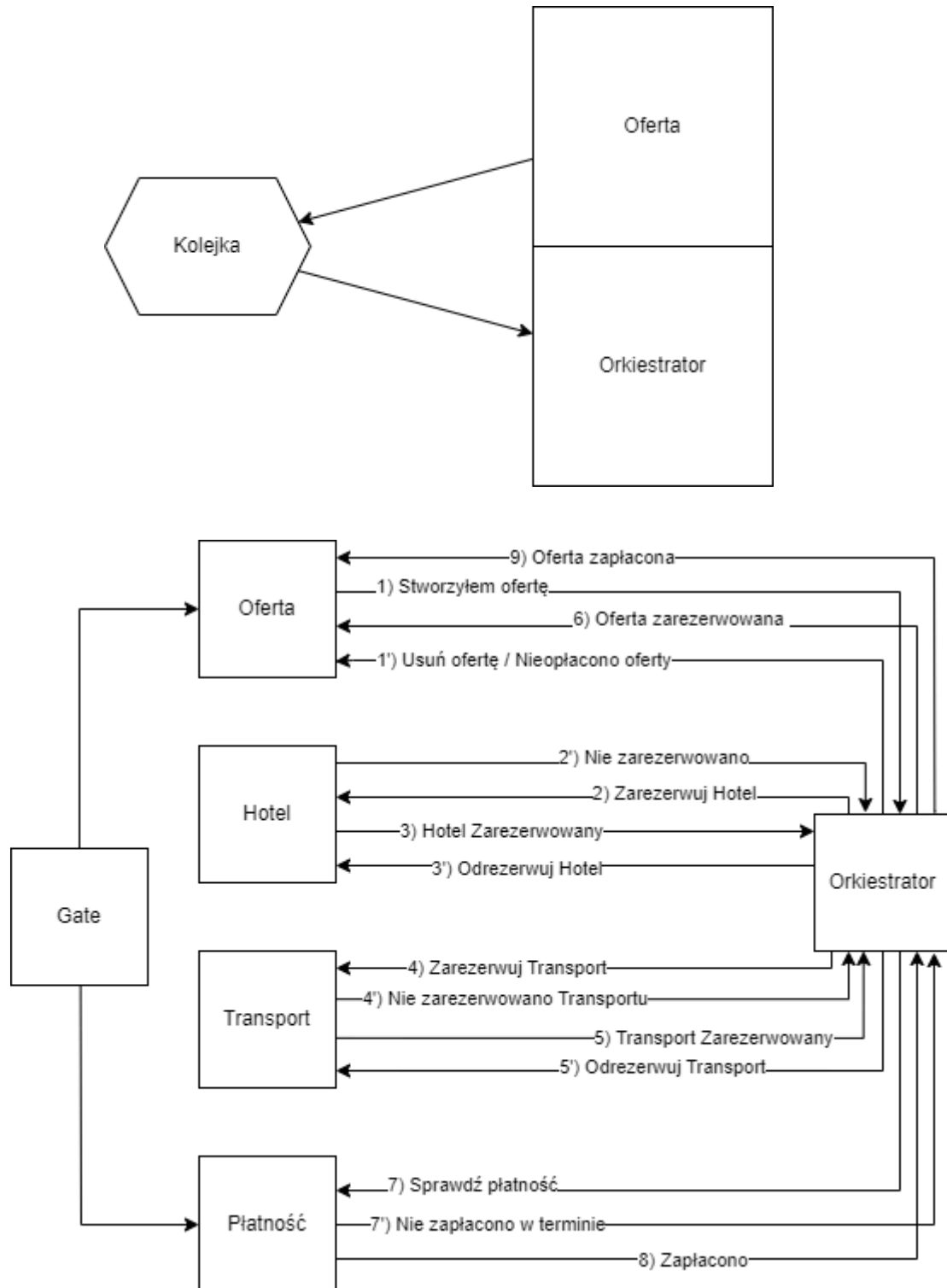


```

PłatnośćDlaOfertyDTO: {
    ZgłoszenieId
    OfertaId
}

```

4. Sagi



5. Opis scrappera

6. Diagram klas i metod