Dzień 1:

Na początek zaczynam od stworzenia wstępnego setupu backendu:

- stworzenie schematu bazy danych(w tym pomoże mi spisanie funkcjonalności, które na ten moment myślę, że wystąpią)

- stworzenie solucji składającej się z 4 projektów: API, Application, Domain, Infrastructure(Clean Architecture)

- przeniesienie encji występujących w bazie danych na klasy będące ich odpowiednikami

- potrzebne konfiguracje i stworzenie migracji

Po tych krokach jesteśmy w momencie gdzie możemy zacząć tworzyć konkretne endpointy.

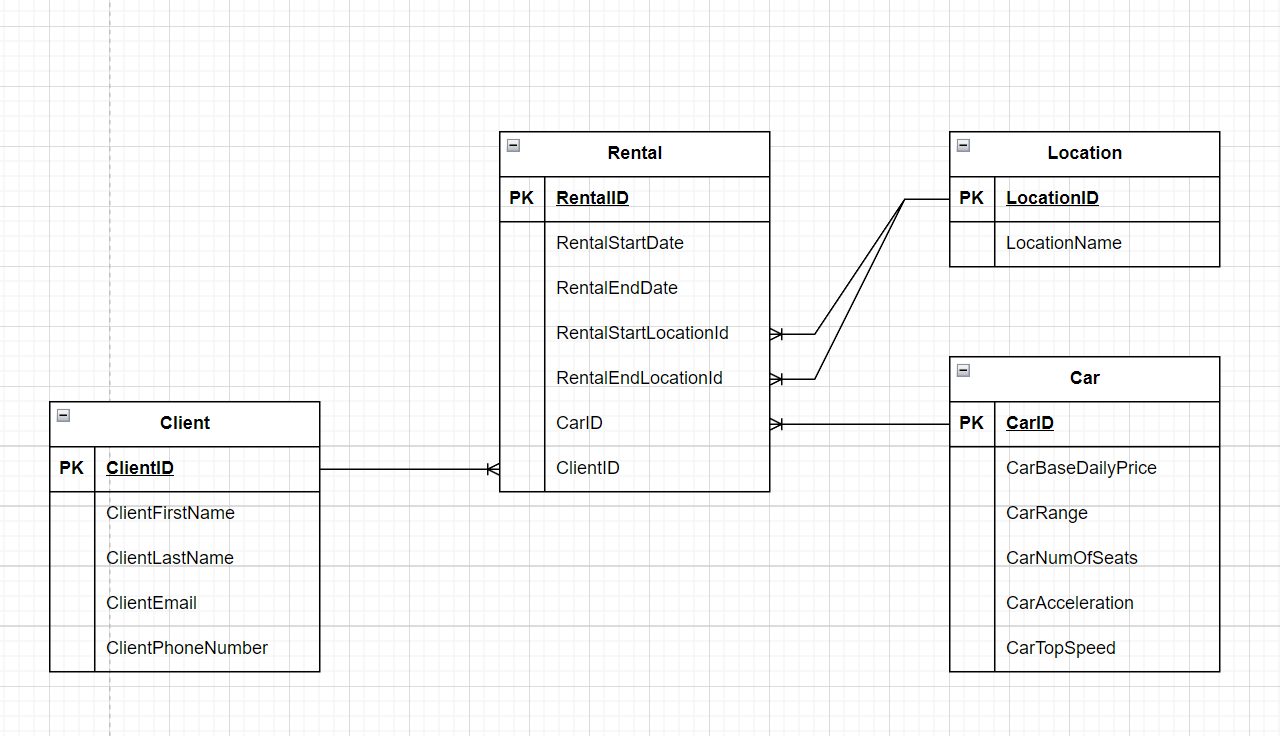
GetLocations()

GetCars()

GetAvailableCars(LocationId, StartDate, EndDate)

GetCarDetails(CarId)

RentCar(StartDate, EndDate, StartLocation, EndLocation, UserDetails)



Tak prezentuje się wstępny schemat bazy.

Pora na pierwsze założenie: nie musimy mieć bezpośredniego powiązania między tabelą Car a Location, ponieważ możemy wywnioskować na bazie tabeli Rental, gdzie w danym momencie znajduje się samochód. Aby wiedzieć w jakim mieście znajduje się samochód na samym początku jego wprowadzenia do bazy, dodamy do tabeli Rental wiersz z jego jednym „sztucznym” wypożyczeniem.

Dzień 2:

Teoretycznie można by dodać do bazy tabelę model ale byłaby zbędna.

Dokończona konfiguracja backendu.

Dzień 3

Napotkałem problem ze zdefiniowaniem API path oraz stwierdzeniem w jakim kontekście mówimy o samochodach dostępnych do wypożyczenia(w końcu nie chodzi nam tyle o samochody co ich wypożyczenie które jest już abstrakcyjnym pojęciem).

Algorytm sprawdzania samochodów dostępnych w danych lokacjach w konkretnych ramach czasowych wymaga pewnych założeń. Dla każdego z pojazdów jego aktualne miejsce pobytu będzie się zmieniało na osi czasu. Załóżmy, że dostępne będą te pojazdy, które w danej lokacji miały koniec swojego wypożyczenia przed wskazaną przez klienta początkową datą jego planowanej rezerwacji, oraz przez ten okres czasu nie ma na nie innych rezerwacji(

Notka do powyższego: jeśli po widocznym ostatnim wypożyczeniu(samochód teoretycznie mógłby być dostępny) dla konkretnego samochodu istnieje wypożyczenie między jego końcem, a naszym wskazanym początkiem oznacza to, że samochód jest niedostępny:

1. Zakończył wypożyczenie i został zwrócony w innym miejscu
2. Wypożyczenie trwa

Chcemy usunąć z listy potencjalnie dostępnych samochody, które spełniają ten warunek).

Po przefiltrowaniu od „lewej strony” rezerwacji pozostało nam sprawdzić czy nie ma innych rezerwacji na ten samochód w trakcie zadanych dat. Jeśli żadna z rezerwacji nie zaczyna się w tym czasie oznacza to, że samochód jest w pełni dostępny ☺

Pan X chce zarezerwować pojazd od 20.03 w Palma Airport i system mu na to pozwala bo ktoś zostawił tam dany model 10.03. System pomyślnie akceptuje operacje. Powinniśmy ustalić dodatkowe filtry, które usuną z listy samochody, które w zadanych ramach czasowych są już zajęte(i to też jest do zrobienia). Problem pojawia się wtedy gdy w dniach 11.03 -19.03 ktoś będzie chciał wypożyczyć dane auto, ale zwróci je już w innej lokacji. I tu wtrąca się czynnik ludzki czyli założenie, że pracownicy firmy w wyznaczonym czasie, bądź przed wypożyczeniem transportują pojazd na miejsce. Mając mało pojazdów taka optymalizacja jest tym bardziej ważna. Nie możemy przewidzieć przyszłości i wiedzieć jak będzie wyglądać lista samochodów dokładnie w danym dniu bo po drodze mogą dochodzić kolejne zamówienia. Najlepszą opcją moim zdaniem jest danie pracownikom możliwości uzupełniania punktów. To znaczy, każdego dnia będą oni mogli zobaczyć gdzie, jakich i ilu samochodów brakuje. W tym celu stworzę symulacyjną stronę wraz z kalendarzem, na której będą oni mogli zatwierdzać przekazanie samochodu z punktu do punktu.

Dzień 4:

Udało się skonstruować działające zapytanie zwracające dostępne samochody.

Trzeba przerobić bazę : nie zwróciłem uwagi na to że samochody są generyczne i nie można na nich określić konkretnych instancji, więc dodaję Car zmieniam na Model. Dodaję Car w innej formie.

Dzień 5:

Stworzony Create dla wypożyczeń z obsługą notyfikacji SMS i Email, wraz z którymi klient będzie otrzymywał numer zamówienia, który będzie mógł wykorzystać na stronie do sprawdzenia szczegółów zamówienia. Od strony BE zrealizowane wszystkie podstawowe założenia. Dopinam front i biorę się za dodatkowe feature`y.

Dzień 6 i 7:

Dodałem kalendarz wraz z wyborem czasu wypożyczenia, który odciąża z konieczności walidacji danych. W przypadku gdy wybrana data początkowa będzie późniejsza od końcowej, kalendarz dla wyboru końcowej zaktualizuje się do tej samej, oraz gdy będziemy chcieli wypożyczyć samochód i zwrócić go tego samego dnia, możemy to zrobić dopiero po godzinie. Po wybraniu dat oraz lokalizacji wyświetli nam się lista samochodów dostępnych dla tych opcji, wraz z ich ilością. Wybieramy jeden z nich i przechodzimy do formularza osobowego.

Dzień 8:

Dodane zostały ficzery dla admin: aktualizacja ceny bazowej, dodawanie nowych samochodów, dla użytkownika: sprawdzanie szczegółów zamówienia na podstawie kodu wypożyczenia.