

Modelowanie i analiza systemów informacyjnych

Projekt - „Dream Cars”

1. Wymagania użytkownika wraz z historią.

Auto-komis aut sportowych „Dream Cars” zgłosił zapotrzebowanie na aplikację, która ułatwi przepływ informacji w firmie, podział zadań, oraz sprzedaż aut klientom.

Firma przechowuje informacje o swoich klientach oraz pracownikach. Dla każdego klienta oraz pracownika odpowiednio, id (pole unikalne), imię, nazwisko, numer dowodu osobistego (pole unikalne), login oraz hasło. Dodatkowa informacja na temat klienta przechowuje dane takie jak numer telefonu i adres a na temat pracownika typ umowy oraz ważność umowy (od kiedy do kiedy, bezterminowa itd.). Każdy zalogowany klient może przeglądać oferty zapewnione przez „Dream Cars”, co więcej może umówić się na spotkanie w celu prezentacji danego auta, dodać lub usunąć samochód do listy aut obserwowanych. Najważniejszą funkcjonalnością z punktu widzenia klienta jest zakup wymarzonego auta. Firma przewiduje zakup danego auta przez klienta z możliwością dostarczenia pod dany adres.

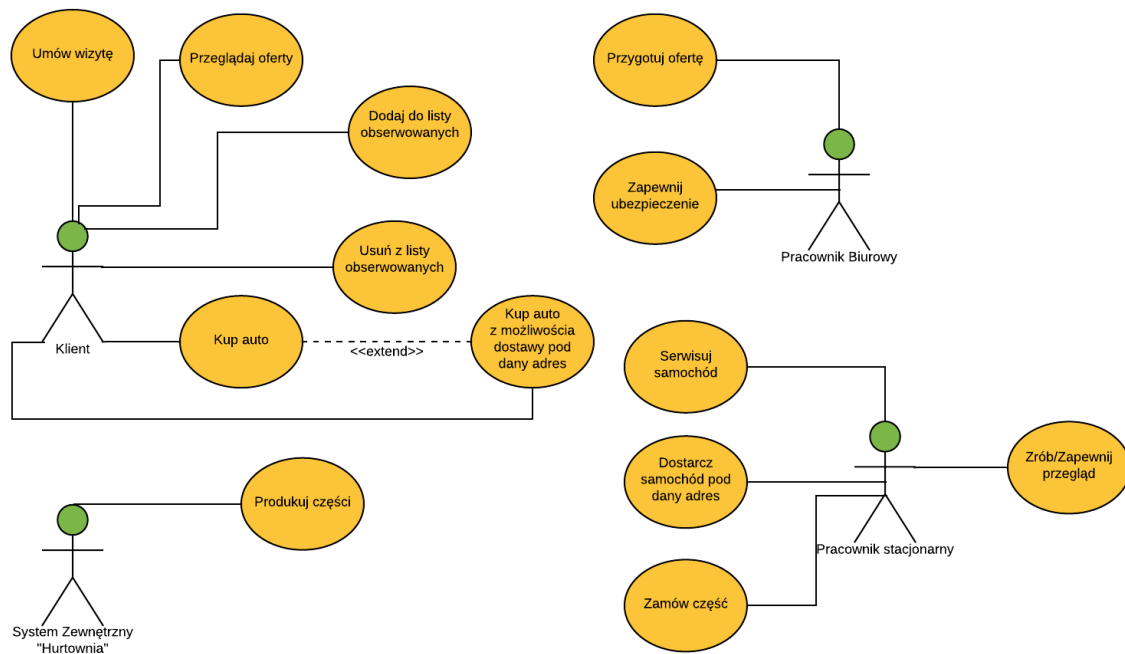
W firmie „Dream Cars” można podzielić pracowników pod względem aspektu: miejsca wykonywanej pracy. Pracownik biurowy przygotowuje oferty odnośnie samochodów oraz pracownik stacjonarny, który opiekuje się samochodem/samochodami. Pracownik stacjonarny może mieć przypisane wiele samochodów do opieki, ale również mogą być pracownicy, którzy nie mają przypisanych samochodów. Co więcej pracownik stacjonarny jest odpowiedzialny za serwis samochodu, zapewnienie przeglądu oraz za dostarczenie auta pod dany adres (w przypadku gdy klient sobie tego zażyczy). Pracownik biurowy jest odpowiedzialny za wystawianie i aktualizowanie ofert odnośnie danego samochodu. Oferta zawiera takie informacje jak id, nazwa, data wystawienia oferty, data zakończenia, cena. Oferta odnosi się do jednego samochodu (może zostać aktualizowana).

Informacje przechowywane w systemie na temat samochodu to: id (pole unikalne), marka, model, generacja, rok produkcji, silnik, pojemność silnika, ilość Koni mechanicznych, numer VIN (pole unikalne, odnosi się do historii auta), oraz informacje czy samochód jest gotowy do sprzedaży czy jest obecnie serwisowany. Samochody dzielą się na auta dopuszczone do jazdy torowej (wyścigi) oraz na auta dopuszczone do ruchu. Dane grupy mogą się przenikać. Odnośnie samochodów dopuszczonych do jazdy torowej musi zostać przechowana informacja odnośnie rozstawu osi (cm), oraz czy samochód posiada klatkę zabezpieczającą. Odnośnie samochodu dopuszczonego do ruchu publicznego musi zostać przechowana informacja odnośnie posiadania świateł przeciw mgiełnych (przedniego oraz tylnego). Każdy samochód posiada swoją historię między innymi informacje takie jak numer „vin”, ilość właścicieli, kraj pochodzenia oraz informacja czy samochód brał udział w wypadku.

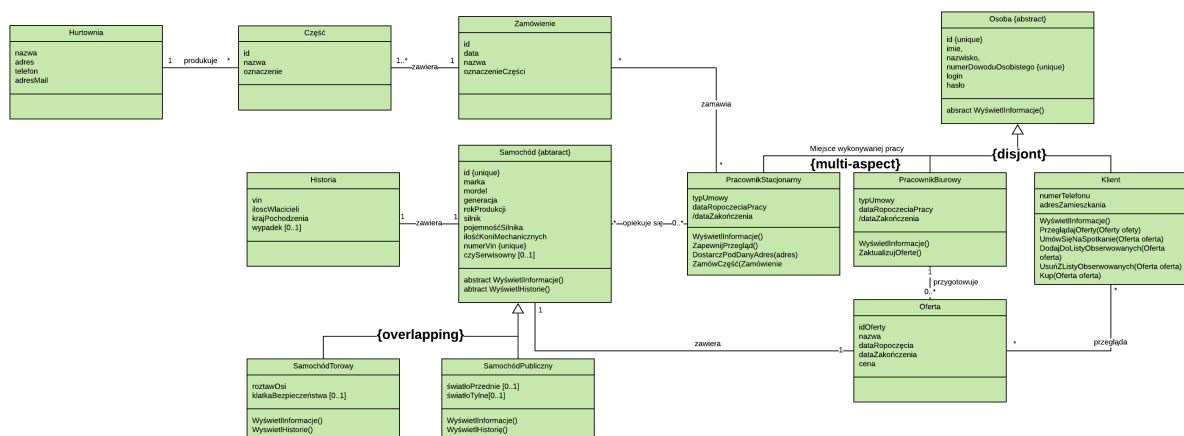
Pracownicy stacjonarni mogą zamawiać odpowiednie części. Zamówienie posiada informacje takie jak id, datę, nazwę, oznaczenie części. Zamówienie może być w odpowiednich stanach

{nowe, w realizacji, odrzucone, zrealizowane}. Część posiada informacje takie jak id, nazwa, oznaczenie. Jedno zamówieni może zawierać wiele części. Części pochodzą z jednej hurtowni, z którą firma ma podpisany kontrakt. O hurtowni przechowujemy takie informacje jak, nazwa, adres, telefon oraz adres email.

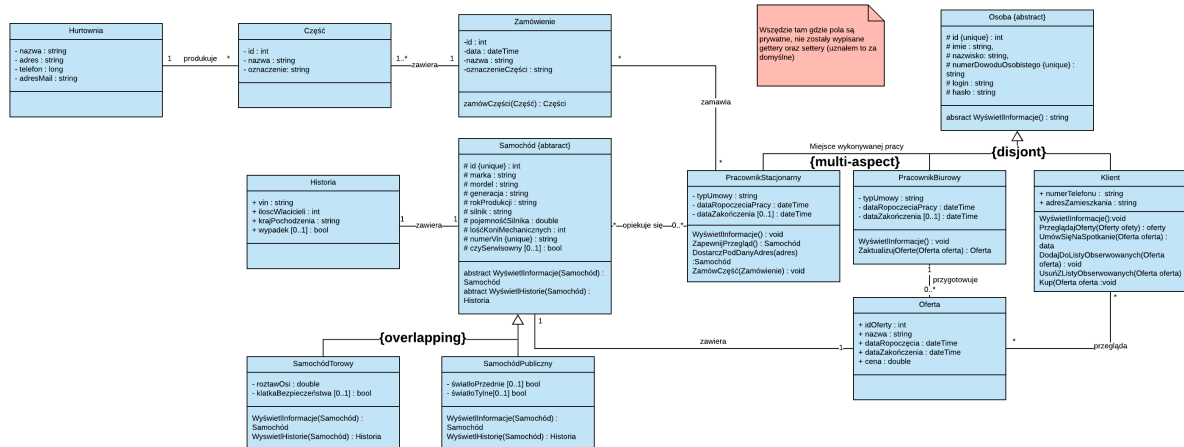
2. Diagram przypadków użycia.



3. Diagram klas – analityczny



4. Diagram klas – projektowy (szczegółowy diagram klas)



5. Scenariusz przypadku użycia

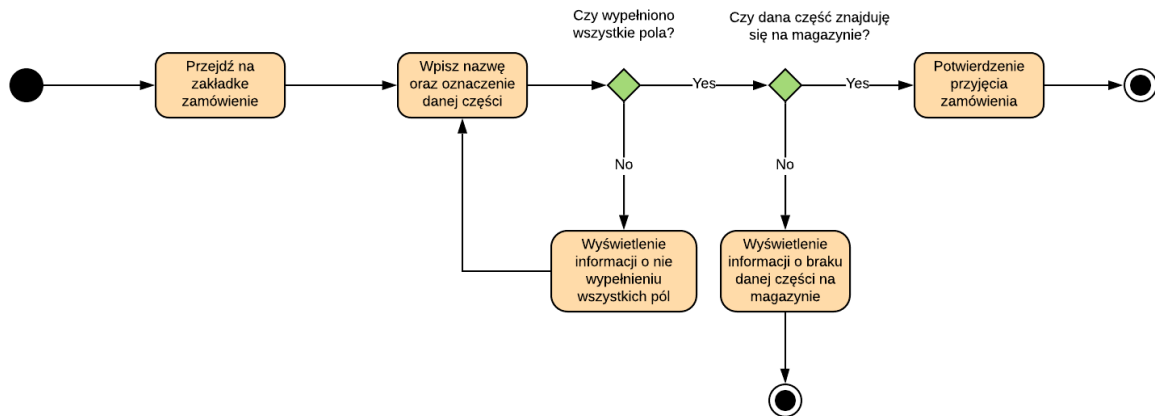
Przebieg zdarzeń między aktorem „Pracownik stacjonarny” a systemem dla przypadku użycia „Zamów część”

1. Przypadek użycia rozpoczyna się, gdy pracownik przejdzie na zakładkę zamówienie.
2. System wyświetla formularz z miejscami na nazwę oraz oznaczenie danej części.
3. Pracownik wpisuje nazwę oraz oznaczenie danej części.
4. System sugeruje („podpowiada użytkownikowi”) nazwę części.
5. Pracownik zatwierdza dane zamówienie odpowiednim przyciskiem.
6. System potwierdza przyjęcie zamówienia.

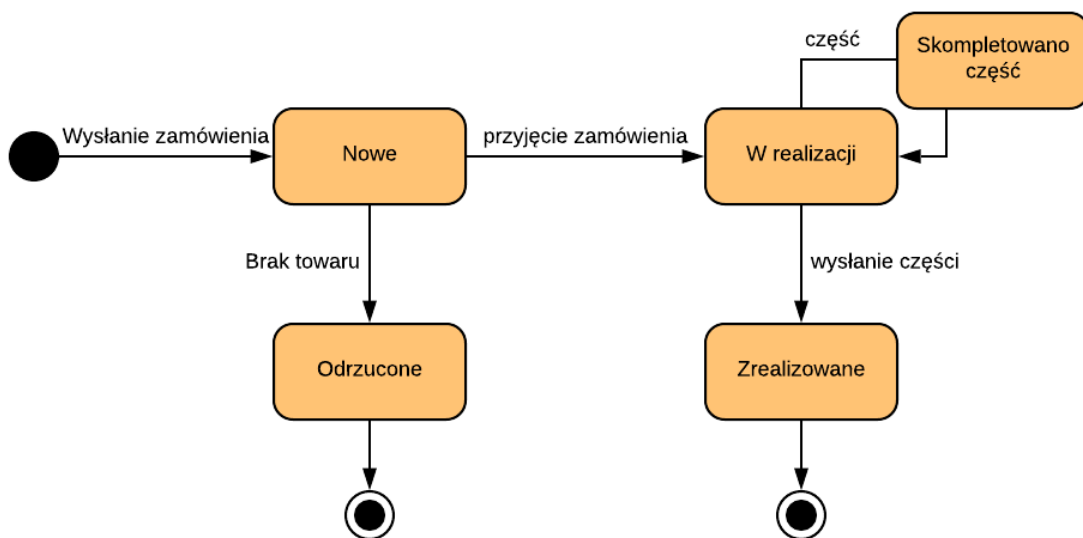
Alternatywny przebieg zdarzeń

- 3a. Pracownik wpisuje nazwę oraz oznaczenie danej części.
- 5a. Pracownik zatwierdza zamówienie
- 6a. System odrzuca zamówienie z powodu braku produkcji danej części przez hurtownię.
- 3b. Pracownik wpisał tylko nazwę bez oznaczenia danej części (lub na odwrót).
- 5b. Pracownik zatwierdza zamówienie.
- 6b. System odrzuca zamówienie z powodu braku wypełnienia odpowiednich pól.

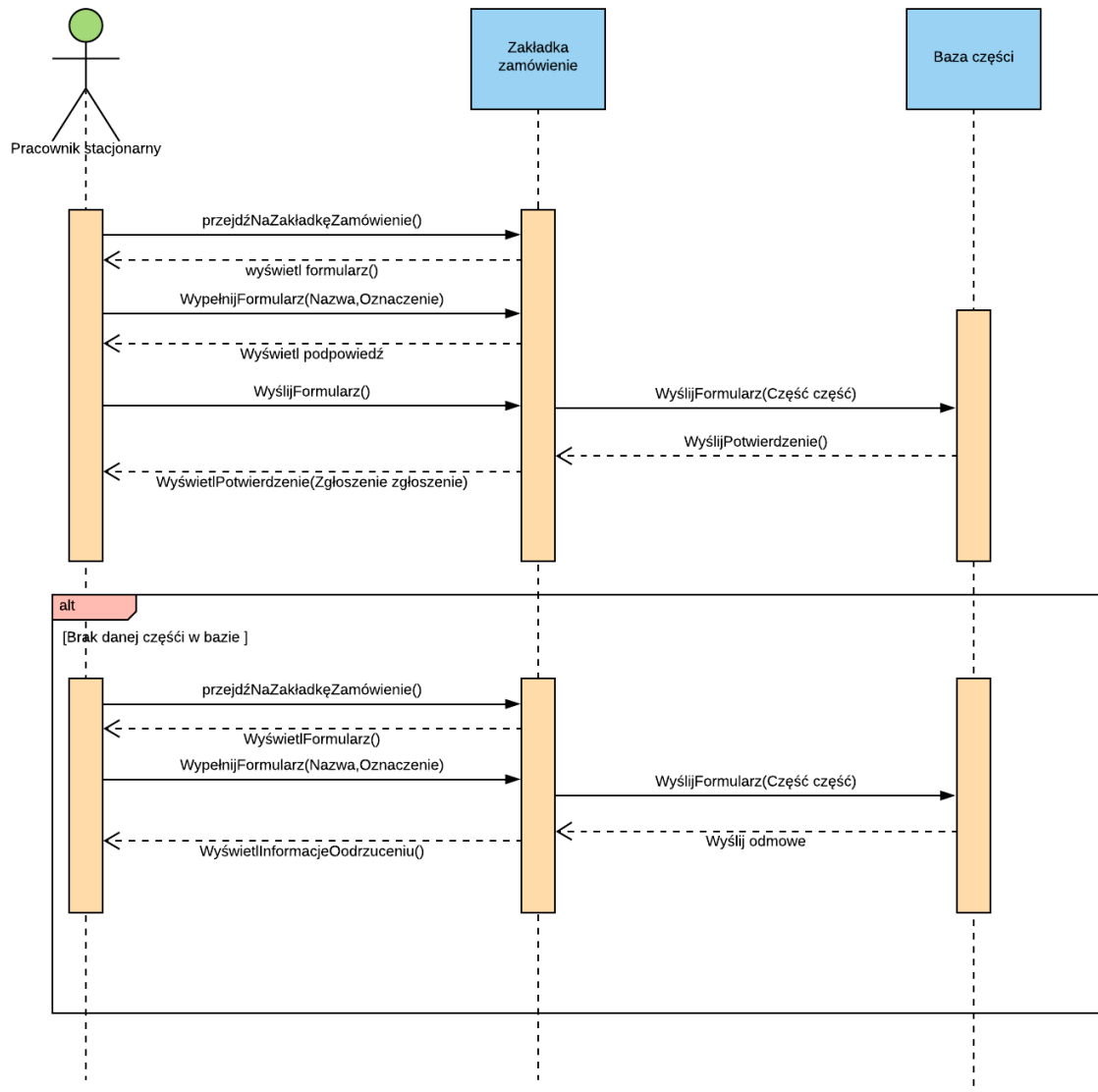
6. Diagram aktywności dla przypadku użycia.



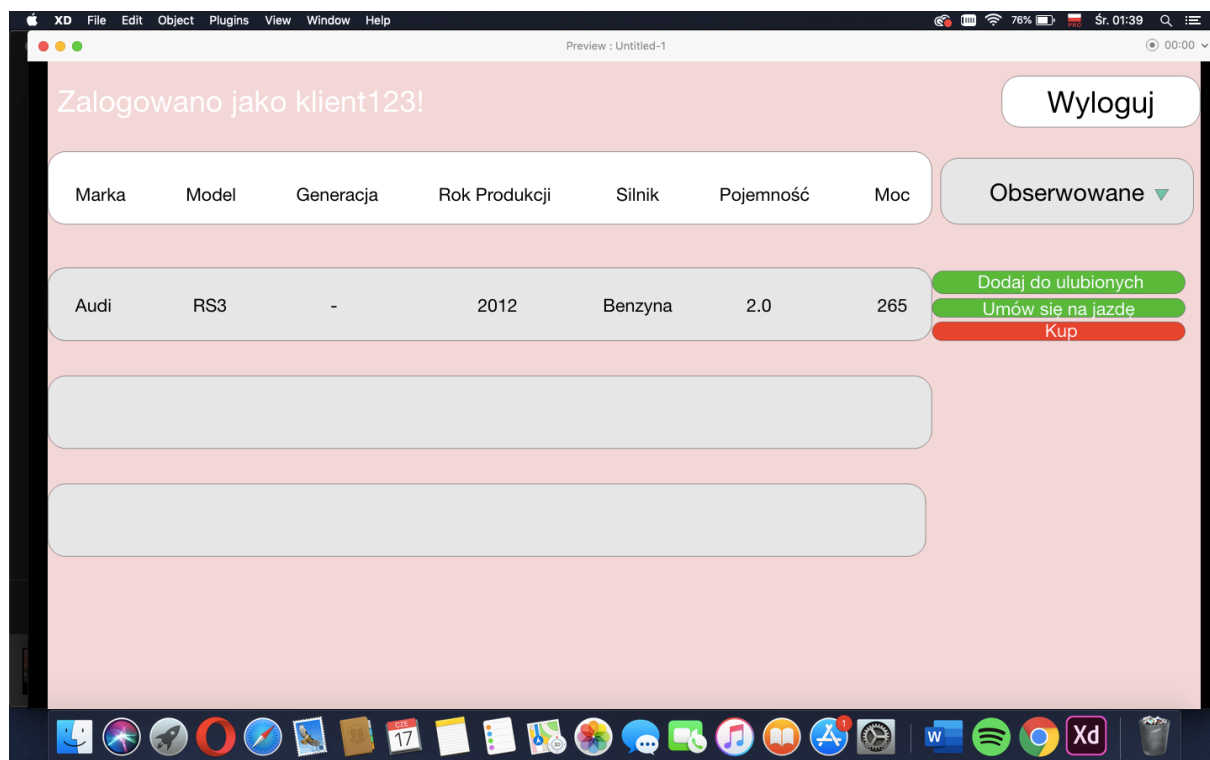
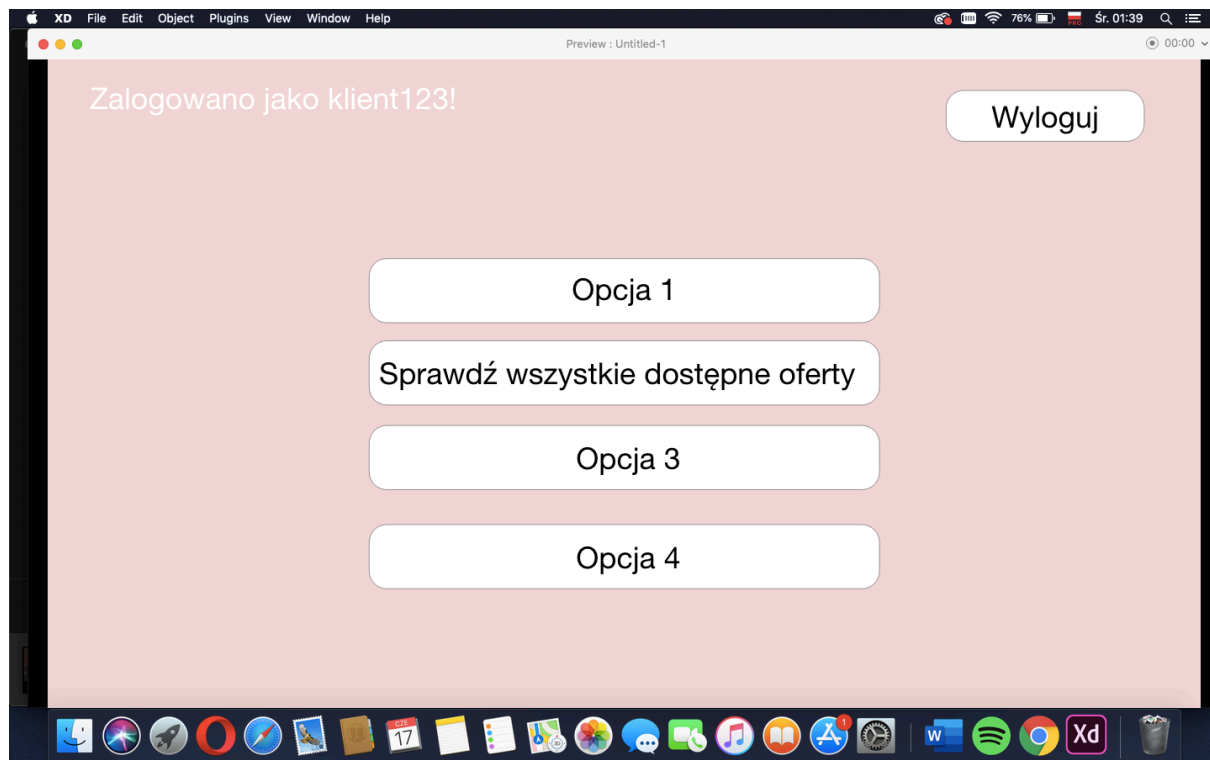
7. Diagram stanu dla klasy – Zamówienie (klasa asocjacyjna między pracownikiem stacjonarnym a częścią).



8. Diagram sekwencji (interakcji) dla danego przypadku użycia („Zamów część”).



9. GUI (dla aktora klient)



10. Omówienie decyzji projektowych i skutków analizy dynamicznej.

Różnice między analitycznym diagramem klas a projektowych polegają na uszczegółowieniu diagramu. W diagramie projektowych musimy pamiętać o typach danych w zależności od języka programowania, o dostępności pól („public”, „private”, „protected”). Musimy pamiętać również, o metodach przypisanych danej klasie, również o ich dostępności, przyjmowanych argumentach czy zwracanym typie danych. W zależności od języka oraz „frameworka” musimy pamiętać o ograniczeniach nadanych przez dany język. Projekt będzie realizowany w języku C#, gdzie nie występuje dziedziczenie wieloaspektowe, dziedziczenie łączne, ograniczenia typu „unique” czy asocjacje „wiele do wiele”. W tym celu w programie zostanie użyty między innymi „EntityFrameworkCore”. Jest to biblioteka, która zapewni nam połączenie z bazą danych (zapewni ekstensje), będzie odpowiadała za połączenie z bazą danych, jak i również za realizację dziedziczenia czy asocjacji. W przypadku dziedziczenia nasza biblioteka, zapewni nam podział na klasy i podklasy wraz z dyskriminatorem, dzięki któremu będziemy mogli określić do jakiej grupy będą należały poszczególne obiekty. Co więcej w przypadku klas asocjacyjnych (asocjacji wiele – wiele) zapewni nam klasę asocjacyjną zgodnie z podejściem realizowanym poprzez aksjomaty relacyjnych baz danych. Zapewnienie ekstensji klas będzie realizowane przez „Microsoft Sql Server”. W analizie dynamicznej musimy pamiętać, że dostępność atrybutów odpowiedniej klasy może się zmienić w zależności od podejścia generowania baz danych. Musimy pamiętać również o realizacji metod, których klient nie przewidział a mogą być używane w przyszłości.